PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 4 : G06F 13/00, 13/14, 13/30

A1

(11) International Publication Number: WO 87/ 00317

(43) International Publication Date: 15 January 1987 (15.01.87)

(21) International Application Number: PCT/US86/01219

(22) International Filing Date: 12 June 1986 (12.06.86)

(31) Priority Application Number: 750,165

(32) Priority Date: 28 June 1985 (28.06.85)

(33) Priority Country: US

(71) Applicant: MICROSCIENCE CORPORATION [US/US]; 8601 Dunwoody Place, Dunwoody, GA 30338 (US).

(72) Inventors: TRAGEN, Barry, J. ; 3014 Forest Gale Drive, Forest Grove, OR 97110 (US). SWIGER, Stephen, M. ; 6215 11th Street, N.E., Portland, OR 97211 (US). GLOVER, William, T. ; 1060 Lakemont Trace, Roswell, GA 30075 (US).

(74) Agent: ZEGEER, Jim; Suite 108, 801 North Pitt Street, Alexandria, VA 22314 (US).

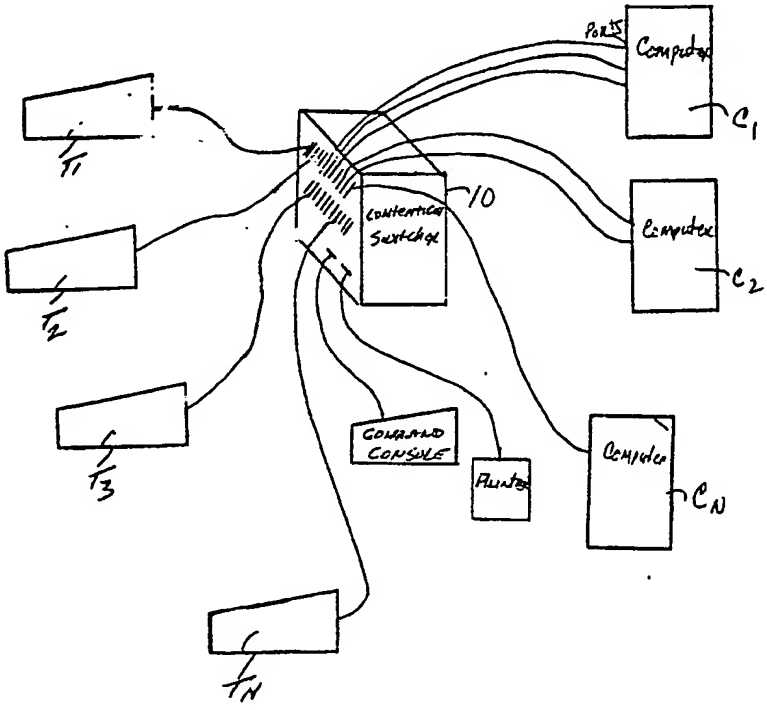
(81) Designated States: AT (European patent), AU, BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), SE (European patent).

Published
With international search report.

(54) Title: CONTENTION SWITCHER

(57) Abstract

Electronic switching system (10) for making logical connections between at least one computer (Cn) and a plurality of computer terminal devices (Tn). Each terminal (Tn) and each computer (Cn) or computer ports is connected to asynchronous receiver-transmitter for converting serial binary data to and from parallel binary data. A microprocessor controls the making of logical connection between each terminal's asynchronous receiver-transmitter and the asynchronous receiver-transmitter connected to a desired or selected computer (Cn). A memory controlled by the microprocessor for storing the identity of computer terminal devices (Tn) and the respective ART to which they are connected, and a connection table of current logical connections between pairs of ARTs. Sequencer logic circuit is controlled by the microprocessor for receiving connection request signals from one of the terminals (Tn) for connection to a specific computer (Cn) and sequentially establishing a logical connection between the ART to which the one terminal (Tn) is connected and the ART to which the specific computer (Cn) is connected.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GA	Gabon	MR	Mauritania
AU	Australia	GB	United Kingdom	MW	Malawi
BB	Barbados	HU	Hungary	NL	Netherlands
BE	Belgium	IT	Italy	NO	Norway
BG	Bulgaria	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SN	Senegal
CH	Switzerland	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
DE	Germany, Federal Republic of	LU	Luxembourg	TG	Togo
DK	Denmark	MC	Monaco	US	United States of America
FI	Finland	MG	Madagascar		
FR	France	ML	Mali		

CONTENTION SWITCHER

BACKGROUND AND BRIEF DESCRIPTION OF THE INVENTION

Contention switching relates to the ability to switch any terminal requesting service into any available port of a "HOST" computer system. A "many-to-few" relationship, much like a rotary telephone system; i.e. fifteen terminals into eight ports. This invention relates to a "logical" switcher as opposed to a "physical switcher". The connection is soft instead of being a metallic path.

If service is unavailable, the user requesting service is automatically queued for service and will be connected automatically when a port becomes available. (Instead of "falling off the end").

Since operating systems for micro-based systems know nothing about switchers, contention switchers incorporating the invention to monitor all traffic across every port to determine when it may disconnect and give the port to another user. Contention switcher incorporating the invention accomodates up to sixteen operating systems simultaneously; watching for trigger messages, and/or "idle" timeouts, and sending appropriate termination sequences. All are user-definable.

All of the above are extended to as many as sixteen "HOSTS" or "GROUPS". While "HOST" generally refers to ports associated with a specific host computer,it is synonomous with

"GROUP", which may include groups of modems or terminals. A further feature of the invention is that any terminal may be designated as a member of a "GROUP", and for each group there is a master. With a few keystrokes, the master may at any time override his group's individual activities, causing their keyboards to be locked and his display to be replicated onto theirs, without their jobs being terminated. Once released by the master, they may resume at the same place. This capability is particularly useful in a teaching environment, and for demonstrations to groups larger than can gather around a single terminal.

Different computer systems have different standards as to data bits, stop bits, and parity. (i.e. ALTOS brand computers require 8 data, 1 stop and no parity, while NCR brand computers require 7 data, 1 stop and even parity.) Terminals connected directly or through a physical switcher would normally require reconfiguration to move from one system to another. A further feature of the invention is that the connection and data transfer is transparent because it is not a metallic path and each port stands alone. Different peripheral devices operate at different speeds of data transfer. (i.e., terminals normally operate at 9600 baud, while modems typically operate at 1200 baud.) This invention allows devices of differing speeds to be logically switched and connected with total transparency. (i.e. the 1200-baud modem can be connected to the 9600-baud terminal and the invention provides buffering and speed translation, within

limits.) Contention switchers incorporating the invention are fully capable of performing data transfer to/from its limit of 128 ports, and all of the above functions, with zero degradation to its users. File transfers are easily and efficiently accomplished through the use of the CW tm package or equivalent. This invention can accomplish all of the functions of a Local Area Network (LAN) with zero degradation at either terminal or processor level.

The basic system consists of thirty-two (32) user ports and two system ports (console and logging printer). A dedicated console device is not required.

A feature of the invention is the system is expandable to a 128 user ports in increments of 32 ports, all field-installable by customer personnel. However, it will be appreciated that the principles and features of the invention are not limited to this number of ports.

Contention switchers incorporating the invention are command driven, even though the resulting action may be an interactive display. Normal use requires as little as a single keystroke (i.e. "2") to request the next port on a specific host system, but other commands are:

SETUP	TERMINAL	CONNECT
STATUS	HOST	DISCONNECT
PORT	GROUP	TIME
		DIAGNOSTIC

All system configuration is accomplished through a very

user-friendly full-screen interactive display, offering soft-setup of each parameter or port attribute. Every port is individually configurable to any format, speed, use, group designation, privacy restriction and host-connect restriction.

A full-screen display may be requested at any time to show current operating configuration and connections, including time-of-day connected, etc. The contention switcher system includes battery backup which will, after a power failure, provide at least sixty (60) hours of retention of key information.

According to the invention, extensive and exhaustive diagnostic programs are held in PROM memory at all times, and run both on-line and off-line to verify integrity of operation, or to point directly to the source of the problem.

Even though there may be as many as twelve events occurring simultaneously, all are executed through Large-Scale-Integration (LSI or VLSI) circuitry and there are no moving parts except for fans.

Power failure is automatically sensed (or may be requested by pressing the logic power switch) and shutdown handled by the invention. When power is again available, the system is automatically brought back up. The Systems Administrator may assign to each terminal port the ability to configure and further to execute diagnostics from that port. Also, the port and its attached terminal may be allowed or denied access to specific hosts or groups.

In the disclosed embodiment, every fourth port has full modem capabilities and may be used to interface to the outside world through modems.

BRIEF DESCRIPTION OF THE DRAWINGS:

The above and other objects, advantages and features of the invention will become more apparent when considered with the following specification and accompanying drawings representing an exemplary embodiment of a contention switching system incorporating the invention wherein:

Fig. 1 is a schematic block diagram of a contention switching system in accordance with the present invention,

Fig. 2 is a schematic block diagram of the central processing unit (CPU) incorporating the invention, (and shown in greater detail in Figs. 8-32),

Fig. 3 is a schematic block diagram of the polling assist hardware components incorporated in the invention,

Fig. 4 is a schematic block diagram of one of the input/output (O/I) boards incorporated in the invention (and shown in greater detail in Fig.s 33-39),

Figs. 5a, b and c are flow charts illustrating the system initialization sequence according to the invention,

Figs. 6a and b are flow charts illustrating the system connect and disconnect sequence according to the invention,

Figs. 7a, b, c, d, e, f, g and h are flow charts

illustrating the various interrupts of the system, and

Figs. 8-39 are detailed circuit diagrams of a contention switcher incorporating the invention.

GLOSSARY - CONTENTION SWITCHER

The following list defines various terms related to the contention switcher which are used throughout this specification.

BAUD RATE-The unit of signaling speed indicating the number of signal transitions per second that occur over a data communications channel.

COMMAND CONSOLE- A data terminal through which the contention switcher is controlled.

CONFIGURATION- The complete specification of all of the necessary parameters required to operate the contention switcher in a specific operating environment.

CONNECTION SEQUENCE- The dialogue entered at a terminal port which causes a logical connection to be made to a host port.

CONNECTION TABLE- An internal list of the current logical connections.

CONTENTION SWITCHER (CTSW)- A single piece of equipment designed to replace the function of many pieces of data communication equipment by creating logical connection paths internally between any two of

its external ports. The contention switcher may be connected to other data communication equipment only if the interconnecting cable has certain signal lines crossed.

CONTROL REGISTER- A special latch used by the microprocessor to control control specific hardware on the CPU board, such as the status indicators and/or the Watchdog Timer.

CPU BOARD- Printed circuit card containing the microprocessor, system memory, time of day clock, programmable timer, sequencer logic, and console/error logger interface.

DATA COMMUNICATION CHANNEL- The entire communications interface, including but not limited to a piece of data terminal equipment connected to data communication equipment which is in turn connected to another piece of data communication equipment and finally terminating at another piece of data terminal equipment.

DATA COMMUNICATION EQUIPMENT (DCE)- A device for the conversion of a serial binary data stream to and from signals suitable for transmission over long distances, specifically telephone lines. Also referred to as a MODEM. The EIA Standard RS-232-C specifies that a female connector is to be associated with DCE.

DATA TERMINAL EQUIPMENT (DTE)- Any device used for the display and transmission of serial data. A computer (being capable of displaying and transmitting serial data) may also be referred to as Data Terminal Equipment. The EIA Standard RS-232-C specifies that a male connector is to be associated with DTE, although this convention is frequently not adhered to.

DIAGNOSTICS- Special commands which are entered at the command console which exercise various portions of the CTSW in order to detect and identify hardware failures.

DISCONNECT TIMEOUT- A period of time after which, if no data activity is detected, a logical connection is broken.

DUAL ASYNCHROUNOUS RECEIVER-TRANSMITTER (DUART) - A signal integrated circuit chip containing all of the digital logic necessary to perform the conversion of serial binary data to and from parallel binary data for two ports. The DUART used in the embodiment is the MC68681.

EIA- Electronic Industries Association
2001 Eye Street, N.W., Washington, D.C. 20006

FIFO- A self-addressed memory chip that remembers the order that data is stored into it, such that the first data written into it is automatically the

first data to be read out of it (First In, First Out).

GROUP- A collection of one or more ports which are handled as a unit. Groups are defined by the GROUP command.

HOST PORT- Any one of the data communication channels which is connected to an external computer port. The connection may be directly to the computer or through a MODEM.

I/O BOARD- Printed circuit card containing sixteen DUARTs and the necessary interface logic required to access those DUARTs from the CPU board. This board also contains the signal drivers and receivers that go to the interface connectors on the back of the contention switcher.

INTERRUPT- An electrical signal generated in order to notify the microprocessor of an external condition that requires the attention of the central processing unit.

INTERRUPT PRIORITY LEVEL (IPL)- The relative importance assigned to each interrupt which indicates to the microprocessor which interrupt to handle first if more than one external device requires it's attention at the same time.

LOGICAL CONNECTION- A connection between two ports of the CTSW. Once a logical connection is made, the two ports

behave as if they are physically connected to each other, although there is a slight delay introduced by the CTSW of less than one character transmission time plus one millisecond.

MASTER TERMINAL- A data terminal that is connected to a port which is configured to control one or more slave terminal ports.

MEGAHERTZ- One million cycles per second.

MODEM (MOdulator/DEModulator)- See DATA COMMUNICATION EQUIPMENT.

MICROPROCESSOR- The central processing unit used to maintain control over all of the internal functions of the contention switcher. The microprocessor used is the MC68010.

NORMAL/SERVICE SWITCH- A two position switch located on the CPU board which is used to lockout certain commands which if inadvertently entered could destroy the configuration.

OPERATIONAL MODE- The normal mode of operation for the CTSW. Certain diagnostic commands which could interfer with normal operating of the CTSW are not allowed to be entered. The opposite of operation mode is shutdown mode.

POLLING- The regular examination by the sequencer of the DUART ports used to transfer data between the ports.

PORT- An electrical passageway into or out of a

computer or a computer related piece of equipment used for data communication. Also used to refer to one of the two data communication channels in a DUART.

PROGRAMMABLE TIMER- An intergrated circuit chip used to generate precisely timed interrupts. The timer used is the MC6840.

REAL TIME CLOCK- An integrated circuit chip used to keep track of the time of day and the current date. Also referred to as "Time of Day Clock".

RS-232-C- An EIA standard defining an "Interface Between Data Terminal Equipment and Data Communication Equipment employing Serial Binary Data Interchange".

SLAVE TERMINAL- A terminal configured to be under the control of a master terminal. When the master terminal enters the override command, the characters entered at the slave terminal port art ignored, and a copy of the data received from the host port connected to the master terminal port is transmitted to the slave terminal.

SHUTDOWN MODE- The mode of operation that the CTSW is required to be in order to run diagnostics. No new connections are allowed to be made at any of the terminal ports.

TERMINAL- See DATA TERMINAL EQUIPMENT.

TERMINAL PORT- Any one of the contention switcher ports which is connected to data terminal equipment. The connection may be directly to a terminal or through a MODEM.

TIME OF DAY CLOCK- See REAL TIME CLOCK.

TRIGGER MESSAGE- A one to sixteen character string, which when received by the CTSW from a host port, initiates a disconnect timeout sequence.

WATCHDOG TIMER- A special programmable timer which, if not regularly attended to by the microprocessor, assumes that the CPU has failed and causes a system reset to occur.

DETAILED DESCRIPTION OF THE INVENTION

Referring to Fig. 1, a microprocessor controlled contention switcher 10 incorporating the invention, has a plurality of ports 11 (typically E1A-Standard RS-232-C) for handling logical connections between one or more computers 12, 13, 14...Cn (sometimes designated host) and multiple peripheral devices such as terminals 15, 16, 17, 18...Tn. According to the invention, if a user of any of the terminal devices wishes to be connected to a particular computer, a request is entered on his terminal. The contention switcher 10 determines if any of the ports of the user selected host computer are available and, if available, achieves a logical connection (as opposed to a

physical switched metallic path or connection) between the requesting terminal and the selected computer. If all of the ports of the selected computer are being used (it being appreciated that the host computer can, in some cases, only have one port) the requesting user will be advised of how many are waiting, if they wish to wait and be queued for service by the selected computer, if that user so desires. Once the logical connection path of a terminal has been made to a selected host computer, the contention switcher monitors all data traffic across every port.

INTRODUCTION TO SYSTEM DESCRIPTION

This specification describes the hardware and software used to implement one embodiment of a contention switcher (CTSW) incorporating the invention. The CTSW is capable of handling up to 64 logical connections between a maximum of 128 EIA Standard RS-232-C devices, at data rates up to 9600 baud.

In the embodiment disclosed herein, the CTSW includes one CPU board and from one to four I/O boards. The CPU 20 used is the MC68010 16-bit microprocessor which is run at 8 megahertz. Each I/O board contains sixteen MC68681 Dual Asynchrounous Receiver/Transmitter (DUART) chips 30. Each DUART 30 is capable of handling two RS-232 ports 31A and 31B, for a total of 32 ports per I/O board. Each port is connected to a terminal (Terminal Port) or to a selected host computer port (Host Port). Both the

CPU board and the I/O boards contain special hardware used to assist the microprocessor 20 in handling the intense data rates required.

The CPU board contains the system memory 35, including up to 8K bytes of battery backed up RAM 35B-1 and 35B-2, used to maintain the system configuration when power is removed from the CTSW. A Time of Day Clock is also maintained by the battery backup system. A programmable timer is used to generate the regular interrupts needed to keep data moving in a timely manner. One DUART on the CPU board handles two RS-232 ports which are not part of the 128 switched ports. These two ports are used to communicate directly with the processor in order to update the system configuration, log connection information, and run diagnostics in the case of a system failure.

A sequencer logic circuit 50 is controlled by a microprocessor 20 for receiving connection request signals from one of the ten . There are two basic modes that the processor operates in to handle DUART I/O. Initially, all ports are handled on a 100% interrupt driven basis. As logical connections are made between ports, they are mapped into a connection table. Periodically, all DUART ports entered into this table are polled in order to do any necessary character data transfers between them. This defines an asymmetrical operation, both interrupt driven and polled, so that initially the CPU can deal with a greater number of interrupts. As more connections are made, less time is available for handling the interrupts, but less time is

needed. The worst case polling loop would consume approximately 50% of the CPU time, leaving approximately 50% for handling interrupts. The best case would be 0% polling time and 100% interrupt handling. The polling loop overhead is totally dependent upon the number of logical connections at any given time.

SYSTEM INTERRUPTS DESCRIPTION

The system processor can be interrupted from any of a number of external events. The following descriptions are ordered from the highest priority event to the lowest. All interrupts (except System Reset) are initially disabled from interrupting the processor until the Interrupt Enable bit is set in the CPU Control Register.

SYSTEM RESET- is generated upon initial power-up of the system or by the RESET push button on the CPU board. A reset is also forced if the watchdog timer enable bit is set in the CPU Control Register and the programmable watchdog timer is allowed to count down to zero.

SYSTEM ABORT-

IPL = 7 is generated by the Abort push button on the CPU board. This interrupt is also caused by a power fail indication from the power supply of a FIFO Full condition from the CPU board interrupt

handling circuitry.

PTM INTERRUPT-

IPL = 6 is generated by the Programmable Timer and is used to initiate polling of the I/O board DUARTs on a regular basis.

RTC INTERRUPT-

IPL = 5 is generated by the Time of Day Clock once each second. It is used to maintain the system clock and to timeout idle system ports.

CPU Board DUART-

IPL = 4 This Interrupt is generated by the DUART on the CPU board.

I/O FIFO Full-

IPL = 3 is generated if on of the FIFOs on any of I/O boards reaches it's maximum capacity of sixteen interrupting DUARTs. This interrupt is used as a load balancing mechanism, giving I/O board(s) that have an excess number of pending interrupts priority over I/O boards that are less busy.

I/O Board DUART-

IPL = 2 This interrupt is generated when any of the I/O Board DUARTs requires attention. These interrupts are presented to the microprocessor in a round-robin fashion by the hardware.

SHUTDOWN SWITCH-

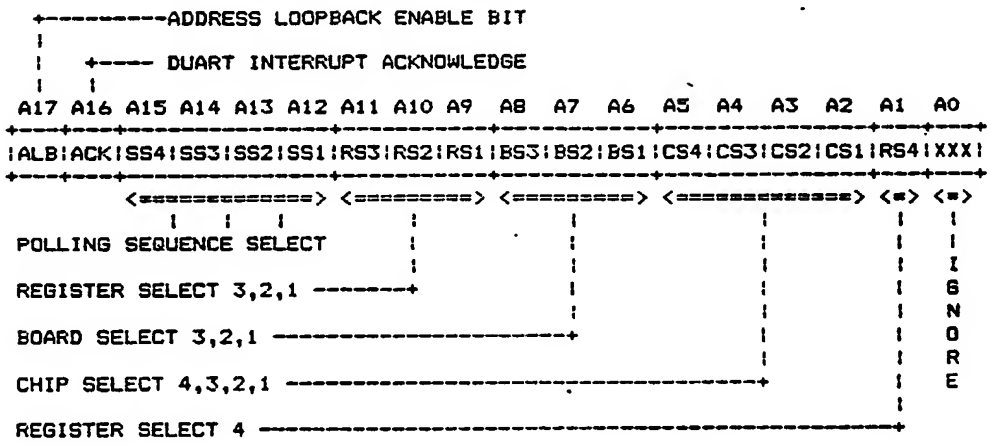
IPL = 1 This interrupt indicates that the front panel

Startup/Shutdown switch is requesting that the system should be shutdown.

COMPREHENSIVE ADDRESS MAP

$FFFFFF $F00000	I/O BOARD DUART ADDRESSING AREA	$FFFFFE $F00001
$EFFFFE $E00000	SEQUENCER RAM DIRECT ADDRESSING AREA	$EFFFFF $E00001
$DFFFFE $D00000	CPU BOARD I/O DEVICE ADDRESSING AREA	$DFFFFF $D00001
$CFFFFF $C00000	CPU BOARD CONTROL REGISTER AREA	$CFFFFF $C00001
	UNUSED ADDRESS SPACE	
$13FFFF $12C000	RESERVED FOR ADDITIONAL WRITABLE ROM (UP TO 128K)	$13FFFF $12C000
$12BFFE $120000	WRITABLE ROM (48K) *** DIAGNOSTIC MODE ONLY ***	$12BFFF $120001
$11FFFE $10C000	RESERVED FOR ADDITIONAL SYSTEM ROM (UP TO 128K)	$11FFFF $10C001
$10BFFE $100000	SYSTEM ROM (48K)	$10BFFF $100001
	UNUSED ADDRESS SPACE	
$03FFFE $032000	RESERVED FOR ADDITIONAL BATTERY BACKED-UP RAM (UP TO 64K)	$03FFFF $032001
$031FFE $030000	BATTERY BACKED-UP RAM (8K)	$031FFF $030001
$02FFFE $005000	RESERVED FOR ADDITIONAL SYSTEM RAM (UP TO 192K)	$02FFFF $005001
$004FFE $000400	SYSTEM RAM (19K)	$004FFF $000401
$0003FE $000000	SYSTEM INTERRUPT VECTORS (1K)	$0003FF $000001

FORMAT OF OFF BOARD DUART REGISTER ACCESS WORD



NOTES:

1) BIT 0 DEFINES EVEN OR ODD BYTE ADDRESSES. DUARTS MAY BE ACCESSED AT EITHER EVEN WORD OR ODD BYTE ADDRESSES.

2) THE POLLING SEQUENCE SELECT IS DECODED AS FOLLOWS:

0000 - ABSOLUTE DUART ACCESS

0001 - POLLING SEQUENCE STEP 1

..... POLLING SEQUENCE .....

0111 - POLLING SEQUENCE STEP 7

11000 - DUART INTERRUPT ACKNOWLEDGE (READ ONLY)

01000 - FIFO FULL INTERRUPT RESET

1001 - DIAGNOSTIC SEQUENCE STEP 1

..... DIAGNOSTIC SEQUENCE .....

1111 - DIAGNOSTIC SEQUENCE STEP 7

ADDRESS LOOPBACK ENABLE IS ONLY VALID FOR DIAGNOSTIC STEPS REFER TO POLLING SEQUENCE STEP DEFINITION FOR MORE DETAIL.

3) THE FOLLOWING TABLE DEFINES THE UTILIZATION OF THE DUART INPUT AND OUTPUT PORT PINS:

OP0 = CTS A **	IP0 = RTS A **
OP2 = DSR A	IP2 = DTR A
OP3 = DSR B	IP3 = DTR B
	IP4 = DCD A **
** = MODEM PORTS ONLY	IP5 = RI A **

SPECIAL PURPOSE CPU BOARD RAM MAP

ADDRESS	UPPER DATA BYTE	LOWER DATA BYTE	
$E041FE		EXPANDED COMPARE RAM	$E041FF
$E04100		OPTION (128 BY 8 BITS)	$E04101
$E040FE		HARDWARE COMPARE RAM	$E040FF
$E04000		(128 BY 8 BITS)	$E04001
$E020FE		HARDWARE POINTER RAM	$E020FF
$E02000		(128 BY 8 BITS)	$E02001
$E000FE		HARDWARE MAPPING RAM	$E000FF
$E00000		(128 BY 8 BITS)	$E00001

NOTES:

1) The Hardware Compare RAM contains up to 8 short messages. Each message can be up to 16 characters in length. Only the low 7 bits are used for the compare operation. The MSB (bit 7) of the character from the Compare RAM indicates the last character of the message. If set, and the final character matches the received character, then Bit 7 of the status byte written into the Pointer RAM is set indicating a complete match. Bit 2 of the Status byte will be cleared, in order to determine if any characters are received after the match occurs.

2) The Hardware Pointer RAM is addressed by bits A7 - A1 of the DUART Register address/command word. During a polling sequence, the Hardware Pointing RAM is used to address the hardware Compare RAM. If the character read from the Compare RAM matches the character read from the DUART then the low 4 bits of the pointer are incremented to point to the next character. Otherwise the low 4 bits of the pointer are reset to point to the beginning of the short message.

ON BOARD I/O DEVICE ADDRESS MAP

ADDRESS	UPPER DATA BYTE	LOWER DATA BYTE	
	MC6840	PROGRAMMABLE TIMER	$DC0FFF
$DC000E		WR TIMER 3/RD LSB BUF	$DC000F
$DC000C		WR MSB BUF/RD TIMER 3	$DC000D
$DC000A		WR TIMER 2/RD LSB BUF	$DC000B
$DC0008		WR MSB BUF/RD TIMER 2	$DC0009
$DC0006		WR TIMER 1/RD LSB BUF	$DC0007
$DC0004		WR MSB BUF/RD TIMER 1	$DC0005
$DC0002		WR CREG 2/READ STATUS	$DC0003
$DC0000		WRITE CONTROL REG 1/3	$DC0001
	MC146818	DATE / TIME CLOCK	$D80FFF
$D8007E		HIGHEST RAM BYTE	$D8007F
$D8001C		LOWEST RAM BYTE	$D8001D
$D8001A		MISC REGISTER D	$D8001B
$D80018		MISC REGISTER C	$D80019
$D80016		MISC REGISTER B	$D80017
$D80014		MISC REGISTER A	$D80015
$D80012		YEAR REGISTER	$D80013
$D80010		MONTH REGISTER	$D80011
$D8000E		DAY OF MONTH REG	$D8000F
$D8000C		DAY OF WEEK REG	$D8000D
$D8000A		HOURS ALARM REG	$D8000B
$D80008		HOURS REGISTER	$D80009
$D80006		MINUTES ALARM REG	$D80007
$D80004		MINUTES REGISTER	$D80005
$D80002		SECONDS ALARM REG	$D80003
$D80000		SECONDS REGISTER	$D80001
	MC68681	CPU BOARD DUART(S)	$D00FFF
$D00E02		STOP CTR/OP BIT RESET	$D00E03
$D00C02		STRT CTR/OP BIT SET	$D00C03
$D00A02		INPUT PORT / OPCR REG	$D00A03
$D00802		INTERRUPT VECTOR REG	$D00803
$D00602		RCVR/XMITTER BUFFER B	$D00603
$D00402		COMMAND REGISTER B	$D00403
$D00202		STATUS / CLK-SEL REG B	$D00203
$D00002		MODE 1/2 REGISTER B	$D00003
$D00E00		CTR / TIMER LOWER REG	$D00E01
$D00C00		CTR / TIMER UPPER REG	$D00C01
$D00A00		INT STATUS / MASK REG	$D00A01
$D00800		IP CHANGE/AUX CTL REG	$D00801
$D00600		RCVR/XMITTER BUFFER A	$D00601
$D00400		COMMAND REGISTER A	$D00401
$D00200		STATUS / CLK-SEL REG A	$D00201
$D00000		MODE 1/2 REGISTER A	$D00001

CPU BOARD I/O REGISTER ADDRESS MAP

ADDRESS	UPPER DATA BYTE	LOWER DATA BYTE	
$CC0000		INT DIAG REGISTER (W/O)	$CC0001
$C80000		STATUS REGISTER (R/O)	$C80001
$C40000		CONTROL REGISTER (R/W)	$C40001

	15 - 8	7	6	5	4	3	2	1	0	
CONTROL REGISTER	NOT USED	LE4	LE3	LE2	LE1	WRE	WPE	WDT	INT	(READ/WRITE)

LE1 THRU LE4 - CPU BOARD LED 1 THRU LED 4, 0 = EXTINGUISH, 1 = ILLUMINATE

WRE - WRITABLE RAM ENABLE/DISABLE CONTROL, 0 = DISABLE, 1 = ENABLE

WPE - WRITE PROTECT ENABLE/DISABLE CONTROL, 0 = DISABLE, 1 = ENABLE

WDT - WATCH DOG TIMER ENABLE/DISABLE CONTROL, 0 = DISABLE, 1 = ENABLE

INT - MASTER INTERRUPT ENABLE/DISABLE CONTROL, 0 = DISABLE, 1 = ENABLE

	15 - 8	7	6	5	4	3	2	1	0	
STATUS REGISTER	NOT USED	N/S	ACF	CJ1	CJ0	WRE	WPT	WDT	INT	(READ ONLY)

WRE,WPT,WDT,INT - IDENTICAL TO CONTROL REGISTER BITS

N/S - NORMAL/SERVICE MODE SWITCH, 0 = NORMAL, 1 = SERVICE

ACF - A/C POWER FAILURE DETECTION, 0 = NOT FAILED, 1 = FAILED

CJ1 AND CJ0 CONFIGURATION JUMPERS, 0 = INSTALLED, 1 = REMOVED

THE CJ2-CJ1 JUMPERS WILL BE INTERPRETED AS FOLLOWS:

BOTH JUMPERS	INSTALLED =	9600 BAUD CONSOLE PORT
CJ1 ONLY	INSTALLED =	1200 BAUD CONSOLE PORT
CJ0 JUMPER	INSTALLED =	300 BAUD CONSOLE PORT
BOTH JUMPERS	REMOVED =	110 BAUD CONSOLE PORT

	15 - 8	7	6	5	4	3	2	1	0	
INTERRUPT DIAGNOSTIC REGISTER	NOT USED	FF3	FF2	FF1	FF0	IP3	IP2	IP1	IP0	(WRITE ONLY)

FF3 THRU FF0 - SIMULATE FIFO FULL, 0 = INTERRUPT, 1 = NO INTERRUPT

IP3 THRU IP0 - SIMULATE INTERRUPT PENDING, 0 = INTERRUPT, 1 = NO INTERRUPT

PROGRAMMABLE ARRAY LOGIC (PAL) BOOLEAN SPECIFICATION

The following specification is used to program the PAL that decodes the memory address into several general categories of access (IOEN, HDRAM, LOCIO, MCSRG, LOCM). This PAL also

indicates an interrupt acknowledge cycle (INTAC) or non-interrupt (FCHI) cycle. It generates the signal ERREN* if a user mode access is attempted with the PROTect bit set in the CPU Control Register.

```
PAL16L8                                          PAL DESIGN SPECIFICATION
PATTERN 02-403BPU1      4/1/84 SUMCHECK = 4248
CATEGORY DECODER            FUSES BLOWN =

      A23       A22       A21       A20       FC0
      FC1       FC2       DAS1      PROT      GND
      /OE       /IOEN     /FCHI     /LOCIO    /LOCM
      /CRGS     /HDRAM    /INTACK   /ERREN    VCC

      IF (OE)    IOEN   =  A23 * A22 * A21 * A20 * FC2 * FCHI

      IF (OE)    FCHI   =  /FC2 + /FC1 + /FC0

      IF (OE)    LOCIO  =  A23 * A22 * /A21 * A20 * FCHI

      IF (OE)    LOCM   =  /A23 * /A22 * /A21 * FCHI

      IF (OE)    CRGS   =  A23 * A22 * /A21 * /A20 * FCHI

      IF (OE)    HDRAM  =  A23 * A22 * A21 * /A20 * FCHI

      IF (OE)    INTACK =  FC2 * FC1 * FC0 * DAS1

      IF (OE)    ERREN  =  /FC2 * PROT
```

PROGRAMMABLE ARRAY LOGIC (PAL) BOOLEAN SPECIFICATION

The following specification is used to program the PAL that decodes the general category of access into several specific enable/clocking signals (DUSEL, RTCSEL, PTMSEL, MCRWR, MSRRD, IDRWR). It also generates Valid Peripheral Address (VPA) back to the microprocessor, and an error signal (ERREN) if an invalid access is attempted.

```
PAL16L8                                              PAL DESIGN SPECIFICATION
PATTERN 04-403BPU4     7/15/84 SUMCHECK = 8308
SELECT DECODER               FUSES BLOWN = 1017

      A19        A18        /ERREN     /VMA       /INTACK
      WRITE      FC2        /LOCIO     /CRGS      GND
      DAS1       /IOERR     /DUSEL     /RTCSEL    /PTMSEL
      MCRWR      /MSRRD     IDRWR      /VPA       VCC

      IF(VCC)  IOERR  = DAS1 * ERREN * LOCIO
                      + DAS1 *          LOCIO * /A19 * A18
                      + VMA *  /FC2  * CRGS
                      + VMA *          CRGS * /A19 * /A18
                      + VMA * WRITE  * CRGS * A19 * /A18
                      + VMA * /WRITE * CRGS * A19 * A18

      IF(VCC)  DUSEL  = DAS1 * /ERREN * LOCIO * /A19 * /A18

      IF(VCC)  RTCSEL = DAS1 * /ERREN * LOCIO * A19 * /A18

      IF(VCC)  PTMSEL = DAS1 * /ERREN * LOCIO * A19 * A18

      IF(VCC)  /MCRWR = /FC2 + /VMA + /CRGS + A19 + /A18 + /WRITE

      IF(VCC)  MSRRD  = FC2 * VMA  * CRGS  * /A19 * A18 * /WRITE
                      + FC2 * VMA  * CRGS  * A19 * /A18 * /WRITE

      IF(VCC)  /IDRWR = /FC2 + /VMA + /CRGS + /A19 + /A18 + /WRITE

      IF(VCC)   VPA   = DAS1 * CRGS
                      + DAS1 * INTACK
                      + DAS1 * LOCIO * A19 * /A18
```

OVERVIEW OF OFFBOARD DUART I/O HANDLING

When a successful connection sequence is entered at a terminal port, the selected host port and the terminal port are logically connected by entering their I/O addresses into a RAM (Mapping RAM) on the CPU board. One bit of these I/O addresses is used to define an optical master/slave relationship between a configurable group of terminals. The master terminal may enter a

command which will cause all slave terminals to be driven with the characters received from the master terminal's host port. Additionally, while these terminals are slaved, no data will be transferred to their respective host ports.

Once a pair of DUART ports have been logically connected, a regularly timed polling sequence conditionally transfers data between them. First, the status of the two ports is latched on the CPU board. If the status latched indicates that the receiver is ready in the source DUART and that the transmitter is ready in the destination DUART then (except for a slaved terminal port) a MOVE instruction will transfer the data from the source DUART receiver to the destination DUART transmitter. If any required condition is not met, then the I/O Enable signal to the VERSABUS will not be asserted (as well as other signals associated with the message compare logic described below) and the MOVE instruction will have no effect on the DUARTs addressed. The MOVE instruction is executed, irregardless of the latched status. Therefore, the MOVE is a hardware conditional, rather than a CPU internal test.

As each character is received from a selected host port, it is compared to one of 8 short messages contained in a RAM (Compare RAM) on the CPU board. As each character of the message is matched, a pointer in another RAM (Pointer RAM) is incremented. If the MSB of the character read from the Compare RAM is set, and that character matches the character being received, a bit is set in a status byte which is written into the

pointer RAM, indicating that the message has been completely matched. This status byte is periodically checked by the processor, and if no characters are received from the terminal port within a configurable timeout period, the connection will be terminated.

Abnormal DUART conditions (Overrun Error, Framing Error, Received Break, Input Change) are always processed by the DUART interrupt handler. I/O board DUART interrupts are queued in a "FIFO" on the I/O board, so that they can be processed on a first come, first serve basis. In order to handle multiple DUART interrupts in an efficient manner, the DUART interrupt handler uses a special address/command word to acknowledge DUART interrupts. After entering the DUART interrupt handler, this address/command word is read to determine the address of interrupting DUART. After handling the interrupt, this address is read again to determine the address of the next DUART to handle. If the address byte read is ZERO, then no more DUART interrupt are pending and the interrupt handler is exited.

POLLING LOOP TIMING

An interrupt from the MC6840 programmable timer initiates a polling loop once every 900 microseconds. The polling loop initializes seven address registers with the address/commands

used to poll each pair of logically connected DUART ports. Then the following sequence of five instructions is executed for each logical connection:

```
TST.W     (A0)+          LATCH TERMINAL PORT DUART ADDRESS
TST.W     (A1)+          LATCH TERMINAL STATUS+HOST DUART ADDRESS
TST.W     (A2)+          LATCH HOST STATUS+COMPARE RAM ADDRESS
MOVE.W    (A3)+,(A4)+    MOVE DATA (HOST TO TERM)+CHECK MESSAGE
MOVE.W    (A5)+,(A6)+    MOVE DATA (TERM TO HOST)+STORE STATUS
```

Word access is used so that at the end of each polling sequence each address/command has been incremented by two. The next polling sequence will then address the next connection pair.

The following table shows the expected execution time necessary to process one complete polling loop for the 8 Megahertz CPU:

FUNCTION	UNIT # OF CYCLES	CYCLES	uSECS
Interrupting the Processor	44	44	5.5
Saving Register Contents	20 + 8 X 8 Registers	84	10.5
Setup for Polling Loop	20 + 8 X 8 Registers	84	10.5
Entering the Polling Loop	14 + 6	20	2.5
Poll each Logical Connection	48 X 64 POLLS MAX	3072	384.0
LOOP instruction Overhead	4 + 10 per 8 POLLS	84	10.5
Restoring Register Contents	20 + 8 X 8 Registers	84	10.5
Exiting from the Interrupt	24	24	3.0
TOTAL MAXIMUM EXECUTION TIME		3496	469.0

POLLING LOOP SEQUENCE DEFINITION

The following table shows the utilization of the latches used to implement the hardware assisted polling sequence. An "X" is placed at each step in the sequence where data is either clocked into a specific latch or a specific latch's output is enabled for a particular use. Reference the BLOCK DIAGRAM OF POLLING ASSIST HARDWARE shown in Fig. 3 for the location of each latch.

STEPS ---- TERMS	LAT 1 CLK	LAT 1 ENA	LAT 2 CLK	LAT 2 ENA	LAT 3 CLK	LAT 4 CLK	LAT 5 CLK	LAT 6 CLK	LAT 6 ENA	LAT 7 CLK	LAT 7 ENA	LAT 8 CLK	LAT 8 ENA	SEQ #
TST (A0)+	X													1
TST (A1)+		X	X		X									2
TST (A2)+				X		X	X							3
MOVE (A3)+,				X				X				X		4
(A4)+		X							X				X	5
MOVE (A5)+,		X								X				6
(A6)+				X							X			7

LATCH 1 - Holds the address of the Terminal Port and Slaved Bit

LATCH 2 - Holds the address of the Host Port and Master Bit

LATCH 3 - Holds the status register from the Terminal Port DUART

LATCH 4 - Holds the status register from the Host Port DUART

LATCH 5 - Holds the pointer used to address the Compare RAM

LATCH 6 - Used to update the Compare RAM address pointer

LATCH 7 - Used to update the Hardware Sequencer Status Byte

LATCH 8 - Holds a character to be output to Slaved Terminal Ports

NOTE: The outputs of latches 3, 4 and 5 are enabled during the entire poling sequence.

The following table shows the utilization of the RAMs used to implement the hardware assisted polling sequence. An "X" is placed at each step in the sequence where data is either read into a specific RAM or a specific RAM's output is enabled for a particular use. Reference the BLOCK DIAGRAM OF POLLING ASSIST HARDWARE for the location of each RAM.

TERMS / STEPS	POINTER RAM ENABLE	COMPARE RAM ENABLE	MAPPING RAM ENABLE	MEMORY ACCESS CYCLE	SEQUENCE BITS 14	13	12	11
TST (A0)+			X	READ	0	0	0	1
TST (A1)+			X	READ	0	0	1	0
TST (A2)+	X			READ	0	0	1	1
MOVE (A3)+,	X	X		READ	0	1	0	0
(A4)+	X			WRITE	0	1	0	1
MOVE (A5)+,	X			READ	0	1	1	0
(A6)+	X			WRITE	0	1	1	1

PROGRAMMABLE ARRAY LOGIC (PAL) BOOLEAN SPECIFICATION

The following specification is used to program the PAL that enables the RAMs and the buffers used to directly access those RAMs that are apart of the polling assist hardware.

This PAL also generates the DUART acknowledge (DUACK) and the FIFO Interrupt Reset (FFRST) signals.

```
PAL16L8                                                PAL DESIGN SPECIFICATION
PATTERN 06-601FPU6     7/15/84 SUMCHECK = 5216
SEQUENCE DECODER 1           BLOWN FUSES = 632

        A15         A14         A13         A12         /IOEN
        /WRITE      /HDRAM      DAS         A16         GND
        DAS2        /MAPENA     /PTRENA     /CMPENA     /B4ENA
        /B3ENA      /B1ENA      /FFRST      /DUACK      VCC

  IF(VCC)   B4ENA = DAS2 * HDRAM * /A15 * /A14 * /A13          ;ADDR $E00XXX

  IF(VCC) MAPENA = DAS * HDRAM * /A15 * /A14 * /A13            ;ADDR $E00XXX
                 + DAS2 * IOEN * /A14 * /A13 * A12             ;S1,D1
                 + DAS2 * IOEN * /A14 * A13 * /A12             ;S2,D2

  IF(VCC)   B1ENA = DAS2 * HDRAM * /A15 * /A14 * A13           ;ADDR $E02XXX

  IF(VCC) PTRENA = DAS * HDRAM * /A15 * /A14 * A13             ;ADDR $E02XXX
                 + DAS2 * IOEN * /A14 * A13 * A12              ;S3,D3
                 + /WRITE * DAS2 * IOEN * A14                  ;S4-S7,D4-D7
                 + WRITE  * DAS  * IOEN * A14                  ;S4-S7,D4-D7

  IF(VCC)   B3ENA = DAS2 * HDRAM * /A15 * A14 * /A13           ;ADDR $E04XXX

  IF(VCC) CMPENA = DAS * HDRAM * /A15 * A14 * /A13             ;ADDR $E04XXX
                 + DAS2 * IOEN * A14 * /A13 * /A12             ;S4-D4

  IF(VCC)   FFRST =  DAS  * IOEN * /A16 * A15 * /A14 * /A13 * /A12  ;SEQ 8

  IF(VCC)   DUACK = /WRITE * IOEN * A16 * A15 * /A14 * /A13 * /A12 ;SEQ 8
```

PROGRAMMABLE ARRAY LOGIC (PAL) BOOLEAN SPECIFICATION

The following specification is used to program the PAL that enables the latches that supply various data during the hardware assisted polling sequence.

This PAL also generates the enables for the buffers used to drive address (B5ENA) and data (B6ENA) out on the VERSABUS, as well as the signal which enables the I/O boards (IOEC).

```
PAL16L8                                                PAL DESIGN SPECIFICATION
PATTERN 07-702FPU7      7/15/84 SUMCHECK = 6E72
SEQUENCE DECODER 2            FUSES BLOWN = 843

      A15         A14         A13         A12         /IOEN
      /WRITE      L1BIT7      TERMRDY     HOSTRDY     GND
      DAS2        /IOEC       /L1ENA      /L2ENA      /L6ENA
      /L7ENA      /L8ENA      /B5ENA      /B6ENA      VCC

  IF(VCC) IOEC = IOEN * /A15 * /A14                          ;DIRECT,S1,S2,S3
                + IOEN * /L1BIT7 * HOSTRDY * /A15 * A14 * /A13      ;S4,S5
                + IOEN * L1BIT7 * HOSTRDY * /A15 * A14 * /A13 * A12  ;S5
                + IOEN * /L1BIT7 * TERMRDY * /A15 * A14 * A13        ;S6,S7
                + IOEN * A15 * /A14 * /A13 * /A12             ;FFRST,DUACK

  IF(VCC) L1ENA = IOEN * A13 * /A12                          ;S2,D2,S6,D6
                + IOEN * A14 * /A13 * A12                    ;S5,D5

  IF(VCC) L2ENA = IOEN * A13 * A12                           ;S3,D3,S7,D7
                + IOEN * A14 * /A13 * /A12                   ;S4,D4

  IF(VCC) L6ENA = DAS2 * IOEN * WRITE * A14 * /A13 * A12     ;S5,D5

  IF(VCC) L7ENA = DAS2 * IOEN * WRITE * A14 * A13 * A12      ;S7,D7

  IF(VCC) L8ENA = DAS2 * IOEN * L1BIT7 * WRITE * /A15 * A14 * /A13 * A12 ;S5
                + DAS2 * IOEN * L1BIT7 * /WRITE * A15 * A14 * /A13 * A12 ;D5
                + DAS2 * IOEN * /WRITE * A15 * /A14 * /A13 * A12        ;D1
                + DAS2 * IOEN * /WRITE * A15 * A14 * /A13 * /A12        ;D4
                + DAS2 * IOEN * /WRITE * A15 * A13            ;D2,D3,D6,D7

  IF(VCC) B5ENA = IOEN * /A14 * /A13 * /A12          ;DIRECT,FFRST,DUACK

  IF(VCC) B6ENA = DAS2 * IOEN * /A14       ;DIRECT,DUACK,FFRST,S1-S3,D1-D3
                + DAS2 * IOEN * A14 * /A13 * /A12                ;S4,D4
                + DAS2 * IOEN * /L1BIT7 * A14 * /A13 * A12       ;S5,D5
                + DAS2 * IOEN * L1BIT7 * /WRITE * A14 * /A13 * A12  ;S5,D5
                + DAS2 * IOEN * A14 * A13                  ;S6-S7,D6-D7
```

PROGRAMMABLE ARRAY LOGIC (PAL) BOOLEAN SPECIFICATION

The following specification is used to program the PAL that clocks data into the latches used to hold various data during the hardware assisted polling sequence.

This PAL also generates the signal CTRCLK* used to clock the 4-bit counter that increments the Compare RAM address pointer.

```
PAL16L8                                              PAL DESIGN SPECIFICATION
PATTERN 08-803FPU8      5/1/84 SUMCHECK = 4292
SEQUENCE DECODER 3           BLOWN FUSES = 519

      A15         A14         A13         A12         /IOEN
      /WRITE      L1BIT7      L2BIT7      HOSTRDY     GND
      DAS         /L1CLK      /L23CLK     /L4CLK      /L5CLK
      /L6CLK      /L7CLK      /L8CLK      /CTRCLK     VCC

  IF(VCC)  L1CLK = DAS * IOEN * /A14 * /A13 * A12                    ;S1,D1

  IF(VCC) L23CLK = DAS * IOEN * /A14 * A13 * /A12                    ;S2,D2

  IF(VCC)  L4CLK = DAS * IOEN * /L1BIT7 * /A14 * A13 * A12           ;S3,D3

  IF(VCC)  L5CLK = DAS * IOEN * /A14 * A13 * A12                     ;S3,D3

  IF(VCC)  L6CLK = DAS * IOEN * A14 * /A13 * /A12                    ;S4,D4

  IF(VCC)  L7CLK = DAS * IOEN * A14 * A13 * /A12                     ;S6,D6

  IF(VCC)  L8CLK = DAS * IOEN * L2BIT7 * HOSTRDY * A14 * /A13 * /A12 ;S4,D4
                 + DAS * IOEN * WRITE * A15 * A14 * A13 * /A12       ;D6 ONLY

  IF(VCC) CTRCLK = DAS * IOEN * /A14 * A13 * A12                     ;S3,D3
                 + DAS * IOEN * /L1BIT7 * HOSTRDY * A14 * /A13 * /A12 ;S4,D4
```

PROGRAMMABLE ARRAY LOGIC (PAL) BOOLEAN SPECIFICATION

The following specification is used to program the PAL that defines the various control signals on the I/O board.

```
PAL16L8                                              PAL DESIGN SPECIFICATION
PATTERN 09-914FPU9      6/15/84 SUMCHECK = 36E2
I/O BOARD DECODER            FUSES BLOWN = 432

      RESET       /DAS        /FFRST      /OBVER      /IOEC
      /DAS2       /DUACK      /ACKIN      /BDENA      GND
      INTV        /CSENA      /DATAENA    /FFSRST     /ACKST
      /ACKOUT     /DTACKST    /DISCON     NC          VCC

  IF(VCC)    CSENA = ACKIN * BDENA * /DISCON * DAS * IOEC * /DUACK * /FFRST

  IF(VCC) DATAENA = ACKIN * BDENA * /DISCON * DAS2 * IOEC * /DUACK * /FFRST
                  + ACKIN * BDENA * /DISCON * DAS2 * IOEC * DUACK * /FFRST *
                                                            OBVER * INTV

  IF(VCC)  FFSRST = ACKIN * BDENA * /DISCON * DAS * IOEC * /DUACK * FFRST
                  + RESET

  IF(VCC)    ACKST = ACKIN * BDENA * /DISCON * DAS * IOEC * DUACK * /FFRST

  IF(VCC)  ACKOUT = ACKIN * /BDENA * IOEC
                  + ACKIN * DISCON

  IF(VCC) DTACKST = ACKIN * BDENA * /DISCON * DAS * IOEC
```

INTRODUCTION TO COMMAND DESCRIPTIONS

The Contention Switcher (CTSW) has many various commands which are used to control the system's operation. The operation of these commands is affected by many different things. These things include from which port the command is entered, which mode of operation the CTSW is currently in, and/or the connection status of any particular port. The commands are intended to be flexible, functional, and consistently structured. Most of the commands can be conveniently abbreviated for ease of use. Options are available for obtaining command syntax information directly from the system.

The Contention Switcher operates in several different modes, depending upon commands entered, battery backup status, and a Normal/Service mode switch located on the front edge of the CPU board. This switch is provided as an extra safeguard against inadvertently destroying any configuration information maintained in the battery backed up portion of the system's memory. Besides the SERVICE mode that this switch defines, the system may be in either the SHUTDOWN or the NORMAL mode of operation. One other mode of operation that is halfway between these two modes is the SHUTDOWN IN PROGRESS mode. When the CTSW is commanded to shutdown either by command, or by the front panel pushbutton, a delay occurs to allow users of the system enough time to cleanly

terminate their connections.

Once the system enters the NORMAL mode of operation, a limited set of commands are available to any terminal that is connected to a contention switcher port. These commands allow a terminal to be able to connect to or disconnect from any one of the other ports in the system according to system configuration information. In the case of a port designated in the configuration as the master over one or more slave terminals, a command is avalable that will cause the slaved terminals to receive a copy of the data being received by the master terminal.

Commands are entered in response to the " > " prompt. The prompt is displayed at any terminal after the item is initially started and any character is typed at the terminal. In the case of a port that has already established a connection, " > " prompt is obtained by entering an attention sequence. The sequence consists of holding the break key down longer than the configured break timeout period, or by entering the number of breaks configured as the minimum break count required for attention before the end of the configured break timeout period. The prompt is also displayed when a configurable TRIGGER message is received by a connected port.

DESCRIPTION OF COMMAND SYNTAX

Commands are entered in response to the " >" prompt. The following notation is used for describing the syntax

of commands. All other symbols are entered exactly as shown.

< > The angular brackets enclose a symbol that is replaced by one of a class of symbols that it represents.

: The vertical bar indicates that a choice is to be made. Only one of the symbols separated by this delimiter is allowed to be entered.

{ } Curly braces enclose syntax that is optional.

{}.. Curly braces follows by an ellipsis enclose syntax that is optional and may be repeated one or more times.

The general format of a command is:

{<cmd>} {/<vname>= <val>{,<val>}... :<val>{,<val>}...:/<opt>}...

where:

<cmd> is the name of the command to be executed. If this field is omitted, it is assumed to be a CONNECT command.

<vname> is the name of a parameter with one or more associated values.

<val> is a numeric or symbolic value associated with a parameter.

<opt> is the name of a parameter without an associated value.

Alphabetical characters may be entered in upper and/or lower case. All commands will be converted to uppercase before interpretation. Each command, option, and/or value name may be abbreviated to its lead ambiguous form.

Values may be entered with or without their preceding value names, however, if the value names are omitted the values must be entered in the exact order shown in each command

description. If a command is entered followed by a "/?" (slash, question mark), all available option and/or value names for that command will be displayed.

INDEX TO COMMAND DESCRIPTIONS

The following commands are only available at the command console, and only if the Normal/Service switch on the CPU board is in the Service position.

DEPOSIT - Alter a memory location

EXAMINE - Examine memory locations

MEMTEST - Exercise memory locations

INITIALIZE - Initialize battery backup memory

The following commands are only available at the command console, and only if the system is in the shutdown mode.

DOWNLOAD - Receive configuration information from a host port.

DUARTTEST - Execute Duart Diagnostic test

RESET - Perform a System Reset

SEQUENCER - Execute Sequencer Diagnostic test

STARTUP - Begin NORMAL operational mode

UPLOAD - Transmit configuration information to a host port.

The following commands are available in both shutdown and/or operational mode at the command console only.

GROUP - Establish a name for a collection of ports.

LOGGER - Start/Stop logging activity for specified ports.

MESSAGE - Establish a symbol name for a string of text.

OVERRIDE -Cause slaved terminals to copy master terminal.

SETUP - Alter configuration parameters of system ports.

SHOW - Display configuration parameters of system ports.

SHUTDOWN -End operational mode.

STATUS -Displays the operational status of system ports.

TIME - Display or set Time of Day clock.

The following commands are available at any port only when the CTSW is in the NORMAL operational mode.

CONNECT - establish a port to port connection

DISCONNECT - break a port to port connection

OVERRIDE - override slaved terminal's connections

CONNECT PORT COMMAND

CONNECT { /TO= } <group> { /FROM= } <term> {/WAIT=} <nwait>

where:

<group> identifies the port or group of ports to connect to.

<term> identifies the port to be connected. This option may only be specified from a privileged port.

<nwait> is the queuing threshold indicator. If the number of other connection requests already waiting for a port from the desired group exceeds this number, then the connection request will not be queued. The default queuing threshold is three connection requests.

The CONNECT command establishes a connection to another port. If the command is entered without specifying any parameters, the port will be connected to one of the default connection ports, if any. If a connection was temporarily suspended as the result of a break attention sequence or a TRIGGER match, the connection will be resumed instead.

When the CONNECT commands specifies connection to a group of ports, the first available port in that group with a configuration compatible with the port to connect will be used as the port to connect to. If all of the available ports in the group are already connected to other ports, then the connection request is placed in a queue, provided that the number of connection requests already queued for this group is less than the threshold specified by the /WAIT parameter.

If there are no ports in the specified group that have a compatible configuration (baud rate, bits per character, etc.) with the port to be connected, then an appropriate error message will be displayed.

If a GROUP name is entered in response to the " >" prompt, it will be interpreted as if the group name had been specified on the "/TO" option of the CONNECT command.

DEPOSIT MEMORY COMMAND

```
DEPOSIT   { /START= } <addr> { /DATA= } <data>
          { /INTERVAL= } <intv>  { /REPEAT= } <rept>
          { /BYTE | /WORD | /LONG }
```

where:

<addr> is up to six hex digits defining the system memory address to modify.

<data> is the data to store at the specified address. An error message is displayed if the data will not fit into the memory entity type selected. A memory entity is either a byte, a word, or a longword of memory.

<intv> is the interval between entities of memory to modify. If specified as zero, the same memory location is repeatedly modified. If specified as one, sequential addresses are modified. If specified to be two every other memory entity will be modified. Et cetera, et cetera.

<rept> is the number of times to repeat the command. If zero is specified, or no value follows the /REPEAT option, the command is repeated indefinitely, or until a break is typed at the console.

/BYTE causes one byte of memory to be modified. This is the default system memory display format.

/WORD causes one word of memory to be modified. Only even addresses may be modified in this mode.

/LONG causes one longword of memory to be modified. Only even addresses may be displayed in this mode.

The DEPOSIT Command modifies a system memory location. This command may only be used if the Normal/Service switch on the

CPU board is in the Service position. The modified memory location is displayed in hexidecimal after the memory location is changed.

DISCONNECT PORT COMMAND

DISCONNECT { /PORT= } <pname>

where:

<pname> identifies the port to be disconnected.

The DISCONNECT command breaks a connection between ports. When this command is entered on the command console, it requires the name of a port to disconnect. If the specified port is not connected, an error will be displayed.

When this command is entered at a terminal port, that port is defined as the port to disconnect and the /PORT option is not allowed.

TEST DUARTS COMMAND

```
DUARTTEST ( /BOARD= ) <board> ( /CHIP= ) <chip>
          ( /TESTS= ) <tnum>(,<tnum>)  ( /REPEAT= ) <passes>
          ( /INTERVAL= ) <intv> ( /WAIT= ) <wait>  ( /LOOP )
```

where:

<board> defines which board to exercise. Board zero refers to the CPU board. I/O boards are numbered from one to four. All of the I/O boards are tested if the BOARD option is not specified.

<chip> identifies a specific DUART chip to exercise. All DUARTs on the selected board are exercised if this option is not specified.

<tnum> identifies a specific desired DUART test to execute. If this option is not specified, then all of the DUART tests will be executed.

<passes> is the number of times to repeat the test. If zero is specified, or no value follows the /REPEAT option, the command is repeated indefinitely, or until a break is typed at the terminal.

<intv> specifies the interval between pass completion messages. If specified as zero, each test identifies itself on a separate line along with a PASS/FAIL indication. If the INTERVAL is specified as a non-zero value, then the test identification will not be produced, unless a test fails. If all of the tests specified pass, then the message that

indicates the number of successfully completed passes is output at the specified interval between pass messages.

<wait> specifies a wait count. A delay of approximately 1.25 microseconds per count is added at various points during the test in order to visually identify certain specific events by observing the LEDs on the boards. A wait count of zero indicates that the test should run at full speed.

/LOOP causes the test to loop on any condition that generates an error. This allows visual display of signals on an oscilloscope.

The DUARTTEST Command exercises one or more DUARTs. The test is divided into three major parts. The first part of the DUART tests exercises one DUART at a time. The second part of the test exercises one I/O board at a time, and the third part of the test exercises all of the I/O boards at the same time, as well as the interrupt handling circuitry on the CPU Board.

The first part of the DUART test is divided into seven tests. Test zero writes and reads back all 256 possible 8-bit data patterns to and from the DUART interrupt vector register. Test one writes and reads back the same data patterns from both mode registers of DUART channel A. Test two checks the mode registers of DUART channel B.

TEST DUARTS COMMAND (continued)

The DUART is then initialized and both DUART channels are setup into the data loopback mode. Test three sends all of the 8-bit data patterns to the channel A transmitter and checks them when they are received at the channel A receiver. Test four tests channel B of the DUART in a similar fashion.

Finally, test five enables DUART interrupts for Channel A and commands the channel A transmitter to START BREAK. When the interrupt is received from the channel A receiver, it is checked in order to verify that it came from the correct DUART. The channel A receiver buffer is checked for the correct break indicator character and the channel A transmitter is commanded to STOP BREAK. When the next CHANGE IN BREAK interrupt is received and verified, of if the interrupt is not received within a fixed timeout period, further interrupts from channel A of the DUART are disabled. Test six is a similar test for channel B of the same DUART.

The second part of the DUART test begins with test seven, which is executed only after all sixteen DUARTS on an I/O board have been tested. Initially, all normal DUART interrupts are prevented from interrupting the CPU by writing to the CPU status register. Then all sixteen DUARTs on the selected I/O board are caused to interrupt. This generates a FIFO FULL condition on the I/O board which causes a FIFO FULL interrupt at the CPU which, after being accounted for, will allow the processor to resume normal DUART interrupt handling. The test will fail if all of

the expected interrupts are not received within a fixed timeout period.

The third part of the DUART test begins with test eight, which executed only after the first two parts of the test are successfully completed on all of the selected I/O boards. This portion of the test simulates the complete operating environment of the contention switcher including the sequencer. The mapping RAM is setup to send the characters received by PORT A of the first DUART to the transmitter of PORT B. The characters looped back from PORT B are then moved by the sequencer to the PORT A transmitter of the next DUART being tested. All of the DUARTs under test are similarly chained together except that the characters received at PORT B of the last DUART in the chain are not moved by the sequencer. The Compare RAM is initialized with 8 strings of 16 descending values. The Pointer RAM pointer for each DUART under test is initialized to point to one of the 8 compare strings. The sequencer status byte for each DUART under test is cleared. A simulated polling sequence is started at regular intervals and the test begins. All 256 character patterns are pushed into the PORT A transmitter of the first DUART and each character is checked as it is received at PORT B of the last DUART. If all of the characters match, and the sequencer status byte for each DUART indicates that the compare string matched, the test reports successful completion.

EXAMINE MEMORY COMMAND

```
EXAMINE    { /START= } <addr> { /COUNT= } <count>
           { /INTERVAL= } <intv>  { /REPEAT= } <rept>
           { /BYTE | /WORD | /LONG }
```

where:

<addr> is the first system memory address to examine and display. The value entered is assumed to be in hexidecimal unless otherwise indicated.

<count> is the total number of memory entitys to examine. A memory entity is either a byte, a word, or a longword of memory. The value entered is assumed to be in decimal unless otherwise indicated.

<intv> is the interval between displayed entities of memory. If specified as zero, the same memory location is repeatedly accessed. If specified as one, sequential accesses are made. If specified to be two every other memory entity will be accessed. Et cetera, et cetera. The value entered is assumed to be in decimal unless otherwise indicated.

<rept> is the number of times to repeat the command. If zero is specified, or no value follows the /REPEAT option, the command is repeated indefinitely, or until a break is typed at the console. The value entered is assumed to be decimal unless otherwise indicated.

/BYTE causes the memory display to be formatted in bytes. This is the default system memory display format.

/WORD causes the memory display to be formatted in words. Only even addresses may be displayed in this mode.

/LONG causes the memory display to be formatted in longwords. Only even addresses may be displayed in this mode.

The EXAMINE Command examines a block of system memory. This command may only be used if the Normal/Service switch on the CPU board is in the Service position. Up to sixteen bytes of memory are displayed on each line, in hexidecimal numbers formatted appropriately for the selected memory entity type. Each displayed line has the ASCII character representation for the data on that line appended to the end of the line. If a byte of data has no corresponding ASCII representation, a period (.) is displayed.

DEFINE GROUP OF PORTS COMMAND

```
GROUP { /NAME= } <group> { /ADD= } <port> { ,<port> }
      { /DELETE }    { /REMOVE= } <port> { ,<port> }
      { /RENAME= } <rname> { /DELETE }
```

where:

<group> identifies the group to be displayed, modified or renamed.

<port> identifies a port or group of ports to be added to or removed from the group.

/DELETE indicates that the group is to be dissolved. All ports are removed from the group and the group name and the memory space allocated for the group is released.

<rname > is a new identification for the same group. Any references to the old group name continue to refer to the

new group.

The GROUP command establishes a collection of ports. If a group name is specified with no other options, then a list of the ports belonging to that group is displayed. Ports may be added to a new group or an already existing group. Ports may only be removed from already existing groups.

If no options are specified on the GROUP command, a list of all of the existing groups are displayed. A maximum of (???) groups may be established at any one time.

Port groups have many useful purposes in the command structure. The most important use of a group is to identify the ports that the CONNECT command will use to satisfy a connection request. Other uses include identifying ports to be logged (LOGGER, command), ports to be slaved to other terminals (OVERRIDE command), and ports to change the configuration of (SETUP command).

INITIALIZE CONFIGURATION MEMORY

INITIALIZE (/ALL : /CONFIG : /SYMBOLS)

where:

/ALL causes all battery backed-up data structures to be initialized.

/CONFIG causes only configuration information to be initialized.

/SYMBOLS causes only symbol information to be initialized.

The INITIALIZE command initializes data structures contained in the battery backup memory. Any previously entered configuration information and/or symbol definitions (groups, messages, etc...) are destroyed. This command may be executed only from the console and only if the normal/service switch on the CPU board is in the SERVICE position.

SYSTEM RESET COMMAND

RESET

This command performs the equivalent function of a system reset. A system reset can be caused by system power up, watchdog timeout, or by pushing the reset button on the front edge of the CPU board. System reset is the only point in time that the SERVICE mode of operation may be entered. If the put into normal operation with the STARTUP command, the SERVICE mode will be terminated, and another system reset is required in order to get back into service mode. The RESET command may only be executed when the system in the SHUTDOWN mode.

LOGGER PORT CONTROL COMMAND

LOGGER { /PORT= } <pname> { /START | /STOP }

<pname > identifies the port or group of ports to be logged.

/START indicates that logging is to begin.

/STOP indicates that logging is to be terminated.

The LOGGER command causes all activity of a port or group of ports to be logged at the logger port. Activity logged includes connections to the port, disconnections of the port and any failures related to the port.

If the LOGGER command is entered without either the /START or the /STOP option, then the logging status of each affected terminal will be toggled to be opposite to its previous state.

TEST MEMORY COMMAND

```
MEMTEST   { /START= } <addr> { /COUNT= } <count>
          { /INTERVAL= } <intv>  { /REPEAT= } <rept>
          { /BYTE ! /WORD ! /LONG }
```

where:

<addr> is up to six hex digits defining the first system memory address to exercise.

<count> is the total number of memory entitys to test. A memory entity is either a byte, a word, or a longword of memory.

<intv> is the interval between tested entities of memory. If specified as zero, the same memory location is repeatedly tested. If specified as one, sequential accesses are made. If specified to be two, every other memory entity will be tested. Et cetera, et cetera.

<rept> is the number of times to repeat the command. If zero is specified, or no value follows the /REPEAT option, the command is repeated indefinitely, or until a break is typed at the console.

/BYTE causes the memory display to be formatted in bytes. This is the default system memory display format.

/WORD causes the memory display to be formatted in words. Only even addresses may be displayed in this mode.

/LONG causes the memory display to be formatted in longwords. Only even addresses may be displayed in this mode.

The MEMTEST Command exercises a block of system memory.

This command may only be used if the Normal/Service switch on the CPU board is in the Service position. The test is divided into three parts.

The first test exercises memory one byte at a time. Various data patterns are written to and read back from each byte of memory in the specified area. If the data read back from a memory location does not verify, the memory address, expected data, and actual data read back are output to the command console.

The second test is similar to the first except that the memory is tested one word at a time instead of one byte at a time. If the start address of the memory area to test is odd, this portion of the memory test will not run. This allows testing of memory areas that are byte-addressable only.

The third test writes a test pattern to each byte of memory in the specified area and then checks other locations within the specified area to insure that they have not been modified. If a location that should not have been modified is changed, the memory address, the modified memory address, and the actual data read back are output to the command console.

DEFINE MESSAGE COMMAND

```
MESSAGE { /NAME= } <name> { /EXIT= } <char> { /RENAME= } <rname>
        { /ASSOCIATE= } <aname> { /DELETE }  { /TRIGGER }
```

where:

<name > identifies the message to be displayed, modified or copied.

< char > identifies the character which will identify the end of the message test. The default message text terminator is a carriage-return character. This option must be used if a multiple line message is to be entered.

< rname > is a new name for the same message. Any references to the old message name continue to refer to the same message text.

aname identifies the name of a trigger message to associate this message with. When a port that uses the associated trigger is disconnected from, this message is transmitted to that port. The specified message should cause the host computer attached to that port to logoff any user logged in at that port, and/or any other desired actions upon disconnect.

/DELETE indicates that the message is no longer required. Any text associated with the message name as well as the memory space allocated for the message name itself is released. If there are any references to the message name to delete, an error message will be displayed and the message will not be deleted.

/TRIGGER indicates that this is a disconnect trigger message. The message text may not exceed 16 characters. A question mark in the message text will match any character in that position.

The MESSAGE command establishes a named text string. This text is then referenced by other commands. The total number

of characters defined by all of the message strings may not exceed (???).

If the /TRIGGER option is specified on the MESSAGE command, then the text associated with the message name is used to trigger a disconnect timeout. If no characters are received from the terminal port within the specified timeout period, after the trigger string has been received from the host port, then the connection between the two ports will be terminated. If there is a message associated this trigger it will be transmitted to the host port as part of the disconnect sequence.

A maximum of eight (8) trigger messages may exist at any one time. Any question mark characters specified in the trigger text will match any single character received from the host port in that character position. The trigger message cannot be longer than a maximum of sixteen (16) characters.

OVERRIDE SLAVE PORTS COMMAND

OVERRIDE { /PORT= } <pname> { /SET | /CLEAR }

where:

< pname > identifies the port or group of ports to be overridden.

/SET indicates that override is to be established.

/CLEAR indicates that override is to be terminated.

The OVERRIDE command causes a port or group of ports temporarily receive a copy of all data received at the terminal

port that is configured as the master for the selected ports. This command may be entered from any terminal, however it will only affect ports which are configured with /MASTER set to the port that the command is entered from.

If the OVERRIDE command is entered without either the /SET or the /CLEAR option, then the override status of each affected terminal will be toggled to be opposite to its previous state.

If no ports are specified on the OVERRIDE command then all ports which are configured with /MASTER set to the port that the command is entered from are affected. If the command is entered from the command console, all ports that have /MASTER set to any port other than NONE will be affected.

TEST DUART SEQUENCER COMMAND

```
SEQUENCER { /START= } <start>    { /END= } <end>
          { /TESTS= } <tnum>{,<tnum>}  { /REPEAT= } <passes>
          { /INTERVAL= } <rept> { /WAIT= } <wait>  { /LOOP }
```

where:

<tnum> identifies a specific desired DUART test to execute. If this option is not specified, then all of the DUART tests will be executed.

<passes> is the number of times to repeat the test. If zero is specified, or no value follows the /REPEAT option, the command is repeated indefinitely, or until a break is typed at the terminal.

<intv> specifies the interval between pass completion messages. If specified as zero, each test identifies itself on a separate line along with a PASS/FAIL indication. If the INTERVAL is specified as a non-zero value, then the test identification will not be produced, unless a test fails. If all of the tests specified pass, then the message that indicates the number of successfully completed passes is output at the specified interval between pass messages.

< wait > specifies a wait count. A delay of approximately 1.25 microseconds per court is added at various points during the test in order to visually identify certain specific events by observing the LEDs on the boards. A wait count of zero indicates that the test should run at full speed.

/LOOP causes the test to loop on any condition that generates

an error. This allows visual display of signals on an oscilloscope.

The SEQUENCER Command exercises the sequencer logic on the CPU board. The test is divided into several steps. The first three steps run memory test routines on the Compare, Mapping and Pointer RAMs. The fourth step writes all 256 possible characters and reads back from the latch used to hold the characters copied to slaved terminals. The fifth and sixth step tests all 128 possible DUART addresses that can be latched into the two DUART address latches. The seventh step writes all 256 possible status values to both DUART port status byte latches and checks the sequencer status byte for correct settings. The eight step checks all 256 character patterns against 256 different character patterns stored in the Compare RAM by using the sequencer comparator and checking the sequencer status byte for correct setting of the message match bit. The ninth step tests that the pointer that is used to address the Compare RAM can be incremented thru all compare RAM addresses and that it resets to the beginning of the message when a character does not match.

SETUP PORT CONFIGURATION COMMAND

```
SETUP { /PORTS= } ALL | <pname>{,<pname>}
      { /BAUD= } 9600 | 7200 | 4800 | 2400 | 2000 | 1800
                 |1200 | 600 | 300 | 150 | 134.5 | 110 | 75
      { /BITS_PER_CHAR= } 5 | 6 | 7 | 8
      { /STOP_BITS= } 1.0 | 1.5 | 2.0
      { /PARITY= }  EVEN | ODD | MARK | SPACE | NONE
      { /TIMEOUT= } <tmo>          { /TRIGGER= } <trmsg> | NONE
      { /WELCOME= } <wmsg> | NONE  { /CONNECT= } <cname>
      { /MASTER= } <mport> | NONE  { /RESTRICT= } <rname> | NONE
      { /PRIVILEGE= }  CONFIG | DIAG | INFO | OPER
      { /DISABLED | /ENABLED }     { /MODEM | /NOMODEM }
      { /STARTED }   { /LOCKED }
```

where:

<pname> is a port identifier or group name identifying the port or group of ports to be configured.

<baud> specifies the baud rate of the specified port or ports. Initially, all baud rates are set to 9600 baud.

/BITS specifies the number of bits per character for the port or group of ports specified. Initially, all ports are set to send and receive 8 bits per character.

/STOP specifies the length of the stop bit used for the port or group of ports specified. Initially, all ports are set to send and receive 1 stop bit per character.

/PARITY specifies the parity type of the specified port or ports. Initially, all ports are set to /PARITY=NONE.

< tmo >specifies the number of minutes of inactivity before an __ prompt is automatically issued. If the prompt is not responded to within 15 seconds a disconnect will occur.

< trig > specifies one of the logoff trigger messages. If connected to a port that has a trigger specified, when the trigger message is received from that port, an __ prompt is automatically issued.

< wmsg > specifies a welcome message. This message is sent to the port when the port is first issued an __ prompt.

< connect > is a port identifier or group name identifying the port or ports this port is connected to by default. Refer to the /START option and/or the CONNECT command for detail.

< master>is a port identifier or group name identifying the port or ports which may override this port's connection. Initially, all ports are /MASTER=NONE.

< rstrct>is a port identifier or group name identifying the port or group of ports that are allowed to connect with this port. Initially, there are no restrictions on any ports.

SETUP PORT CONFIGURATION COMMAND (continued)

/PRIVILEGE indicates which kind of commands are to be allowed to be executed at the port or ports being configured.

/DISABLE marks the port or group of ports as unavailable.

/ENABLE marks the port or group of ports as available. All responding ports are initially enabled.

/MODEM identifies the port or group of ports as being connected through a modem. This setting is only valid for the first out of each four ports.

/NOMODEM identifies the port or group of ports as being directly connected. This is the initial setting for all ports.

/STARTED identifies the port or group of ports to be automatically connected upon system startup. The port will be connected to one of the ports defined by the /CONNECT option.

/LOCKED identifies the port or group of ports as being permanently connected. Prompts, timeouts and triggers are disabled for each port with the /LOCKED attribute.

The SETUP command is used to alter or display the configuration parameters of one or more ports. If the SETUP command is entered with no other options, the configuration of all of the ports is displayed. If the name of a port or group of ports is the only supplied parameter, then the configuration parameters are displayed for those ports only.

Alteration of the configurable parameters for a port will only affect the port upon reinitialization of that port. All ports are reinitialized when the system is initially started.

When a port is disabled, no new connections are allowed to be made to or from that port. Any existing connection remains valid until terminated in the normal fashion. Connections may be forced from any privileged port.

If a trigger message is specified for a port, then during any connection to that port, the data stream received from that port is continuously monitored for the specified message. Whenever the specified message is received from the triggered port, a timeout sequence is started for that connection. If no characters are sent to the triggered port before the end of the specified timeout period, the connection will be terminated.

SHOW SHOW PORT CONFIGURATION COMMAND SHOW

```
SHOW  { /GROUP= } ALL | <group>{,<group>}
      { /BAUD= } 9600 | 7200 | 4800 | 2400 | 2000 | 1800
                 |1200 | 600 | 300 | 150 | 134.5 | 110 | 75
      { /BITS_PER_CHAR= } 5 | 6 | 7 | 8
      { /STOP_BITS= } 1.0 | 1.5 | 2.0
      { /PARITY= }  EVEN | ODD | MARK | SPACE | NONE
      { /TIMEOUT= } <tmo>          { /TRIGGER= } <trmsg> | NONE
      { /WELCOME= } <wmsg> | NONE  { /CONNECT= } <cname>
      { /MASTER= } <mport> | NONE  { /RESTRICT= } <rname> | NONE
      { /PRIVILEGE= }  CONFIG | DIAG | INFO | OPER
      { /DISABLED | /ENABLED }      { /MODEM | /NOMODEM }
      { /STARTED }   { /LOCKED }    { /ZPRIV }
```

where:

< group > is a port identifier or group name identifying the port or group of ports to show the configuration of.

< baud > restricts the command to display only those ports which are configured with the specified baud rate.

/BITS restricts the command to display only those ports which are configured with the specified number of BITS per character.

/STOP restricts the command to display only those ports which are configured with the specified stop bit length.

/PARITY restricts the command to display only those ports which are configured with the specified /PARITY setting.

< tmo>restricts the command to display only those ports which are configured with the specified TIMEOUT period.

< trig> restricts the command to display only those ports which are configured with the specified TRIGGER message.

< wmsg> restricts the command to display only those ports which are configured with the specified WELCOME message.

< master>restricts the command to display only those ports which are configured with the specified master ports.

< rstrct>restricts the command to display only those ports which have the specified connection restrictions.

/PRIVILEGE restricts the command to display only those ports which have the specified set of privileges.

SHOW PORT CONFIGURATION COMMAND

/ENABLE restricts the command to display only those ports which are enabled.

/DISABLE restricts the command to display only those ports which are disabled.

/MODEM restricts the command to display only those ports which are configured as being connected to a MODEM.

/NOMODEM restricts the command to display only those ports which are configured as not being connected to a MODEM.

/STARTED restricts the command to display only those ports which are configured to be connected upon system startup.

/LOCKED restricts the command to display only those ports which are configured such that their connections are LOCKED.

The SHOW command is used to display the configuration parameters of one or more ports. If the SHOW command is entered without parameters, the configuration of the port that the command was entered from is displayed.

Specification of any of the configurable parameters will restrict the display to only those ports that are configured as per the parameters specified.

START OPERATIONAL MODE COMMAND

STARTUP

The STARTUP command causes the contention switcher to enter the operational mode. Diagnostic commands are disabled, permanent and/or temporary connections defined in the configuration tables are established, and all other enabled terminal ports are allowed make connections. This command may only be entered at the command console.

END OPERATIONAL MODE COMMAND

SHUTDOWN (/WAIT=) < wait > (/MESSAGE=) < mess >

where:

< wait > is the number of minutes to wait before shutting down. The default shutdown delay is 5 minutes.

< mess > specifies a message which will be broadcast to all connected terminal ports, indicating the reason for shutdown or other appropriate message.

The SHUTDOWN command causes the contention switcher to exit the operational mode. Initially, all terminal ports are disabled from establishing new connections. After the specified waiting period, all logical connections are broken and the CTSW enters the shutdown mode.

DISPLAY PORT STATUS COMMAND

STATUS [/GROUP=] ALL : < group > [, <group>]

[/QUEUES] [/SYMBOLS]

where:

< group > displays information about the current state of the selected port or group of ports.

/QUEUES displays information about the current state of the various systems queues.

/SYMBOLS displays information about the system symbol table.

The STATUS command is used to display information about the operating state of the system. If the STATUS command is entered without specifying any parameters, all available status information is displayed.

DISPLAY/SET TIME OF DAY COMMAND

TIME [/SET]

where:

/SET indicates that time and date are to be set.

The TIME command displays the system date and time. This command may be entered from any terminal port.

If the TIME command is entered with the /SET option then the current date and time is prompted for. The /SET option is only allowed from the command console port.

CLAIMS

1. Electronic switching system for making logical connections between at least one computer and a plurality of computer terminal devices comprising,

a plurality of asynchronous receiver-transmitters (ART) for converting serial binary data to and from parallel binary data, one ART for each said at least one computer and each of said plurality of computer terminal devices, respectively, and means for connecting each respective said computer terminal devices and computer to an ART, respectively,

a microprocessor,

memory means controlled by said microprocessor for storing the identity of computer terminal devices and, said at least one computer and the respective ART to which they are connected and a connection table of current logical connections between pairs of said ARTS,

sequencer logic means controlled by said microprocessor for receiving connection request signals from one of said terminals for connection to a specific computer and sequentially establishing a logical connection between the ART to which said one of said terminals is connected and the ART to which said specific computer is connected.

2. Electronic switching system as defined in claim 1 wherein there are a plurality of ports on said one or more

computers, each of said ports being connected to one of said ARTS, respectively.

3. Electronic switching system as defined in claim 1 wherein there are a plurality of said computers, each having one or more ports thereon, and means connecting each said port to one of said ARTS, respectively.

4. Electronic switching system ass defined in claim 1 including means for monitoring and comparing information data exchanged between logically connected ART pairs to detect hiatus in data transfer for a predetermined period of time, and generating a disconnect signal for said logically connected ART pairs.

5. Electronic switching system as defined in claim 4 including self-addressed memory means controlled by said microprocessor for storing requests from terminals for connection to selected ARTs, respectively.

6. Electronic switching system as defined in claim 1 wherein said sequencer logic means includes a plurality of programmed array logic (PAL) circuit chips.

7. Electronic switching system as defined in claim 1 including at least one further ART connected solely to said microprocessor for entering and retrieving data therefrom.

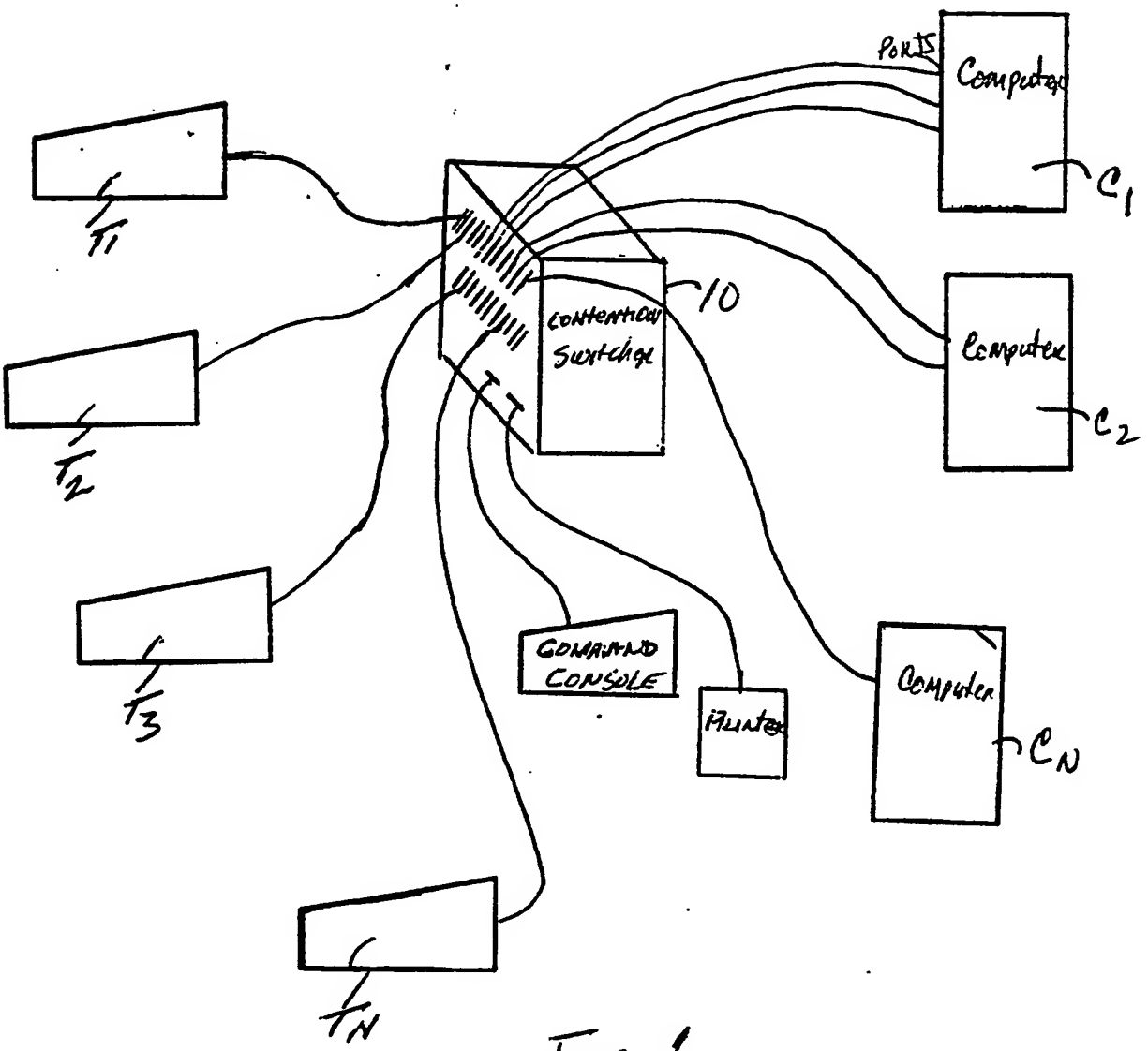

PORTS
Computer
C_1
10
Computer
C_2
COMMAND CONSOLE
PRINTER
Computer
C_N
T_1
T_2
T_3
T_N

FIG. 1

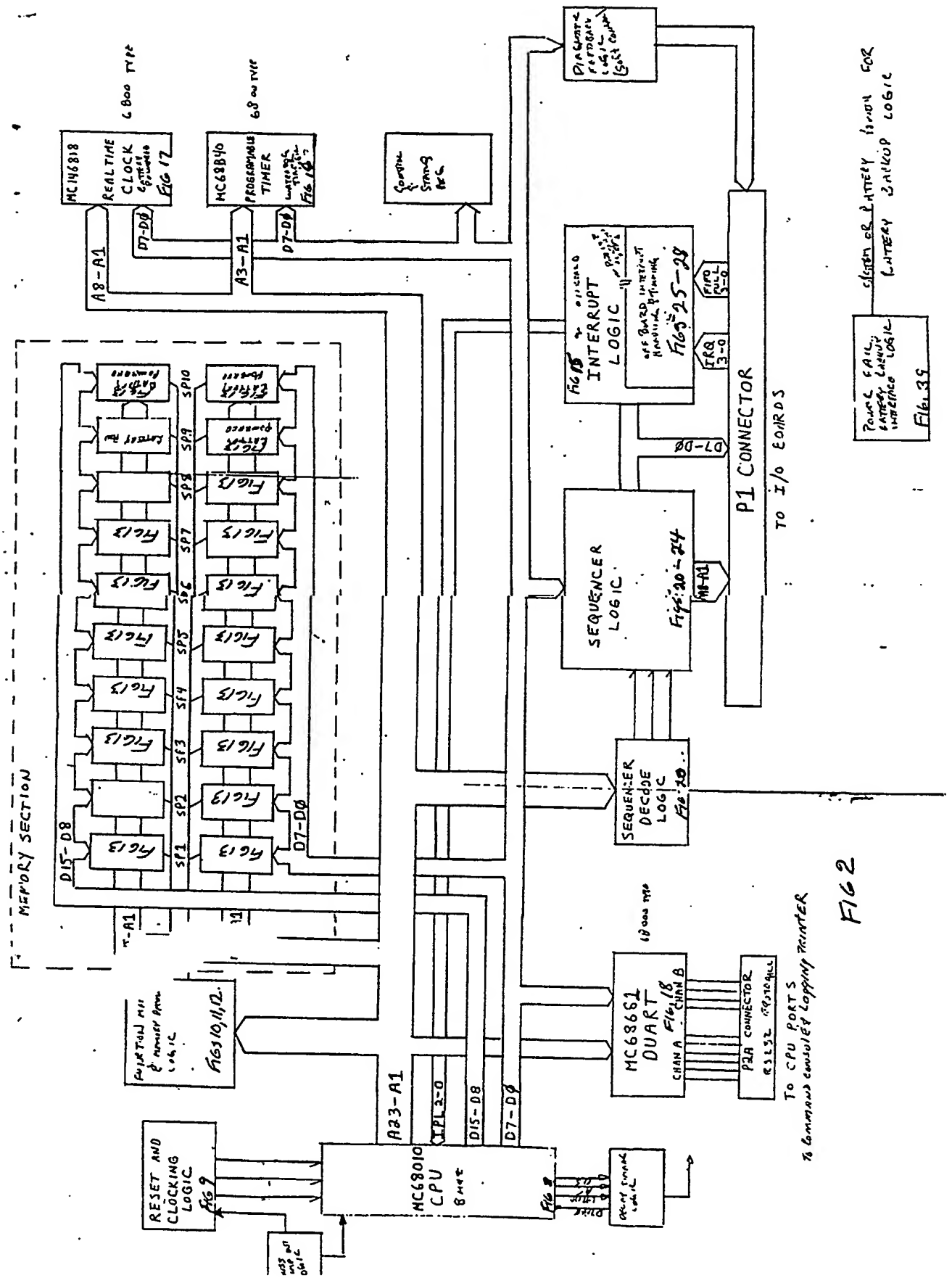

MEMORY SECTION
D15-D8
D7-D0
A-A1
SP1
SP2
SP3
SP4
SP5
SP6
SP7
SP8
SP9
SP10
MC146818
REAL TIME CLOCK
MC68B40
PROGRAMMABLE TIMER
INTERRUPT LOGIC
SEQUENCER LOGIC
SEQUENCER DECODE LOGIC
P1 CONNECTOR
TO I/O BOARDS
MC68010 CPU
RESET AND CLOCKING LOGIC
MC68661 DUART
CHAN A
CHAN B
P2A CONNECTOR
RS232
A23-A1
IPL2-0
D15-D8
D7-D0
To CPU PORTS
& Command controlled Logging Printer

FIG 2

BLOCK DIAGRAM OF POLLING ASSIST HARDWARE

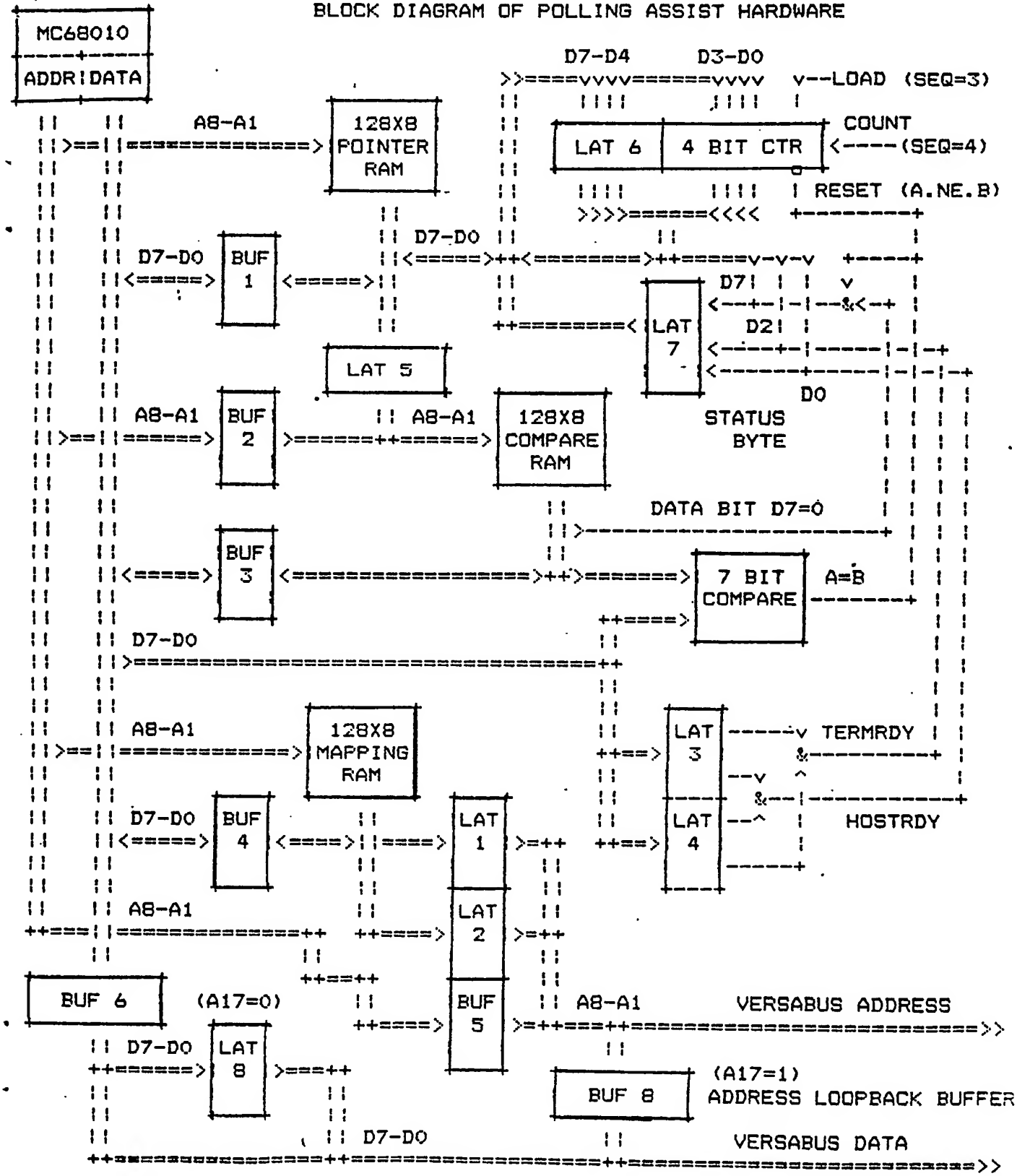


Fig 3

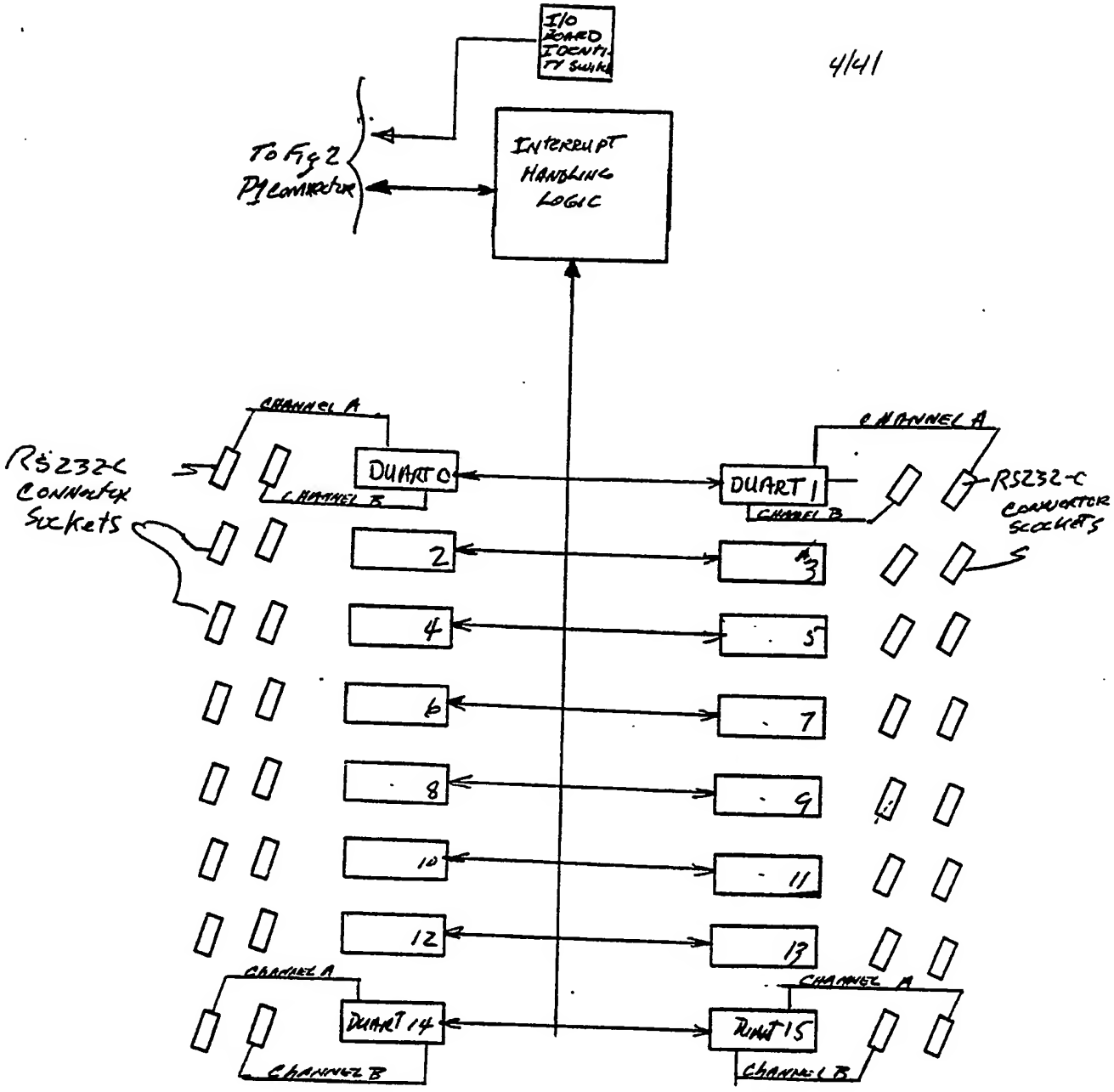

I/O BOARD IDENTITY SWITCH
TO FIG 2 PC CONNECTOR
INTERRUPT HANDLING LOGIC
CHANNEL A
RS232C CONNECTOR SOCKETS
DUART 0
CHANNEL B
DUART 1
RS232-C CONNECTOR SOCKETS
2
4
6
8
10
12
DUART 14
DUART 15

FIG 4.

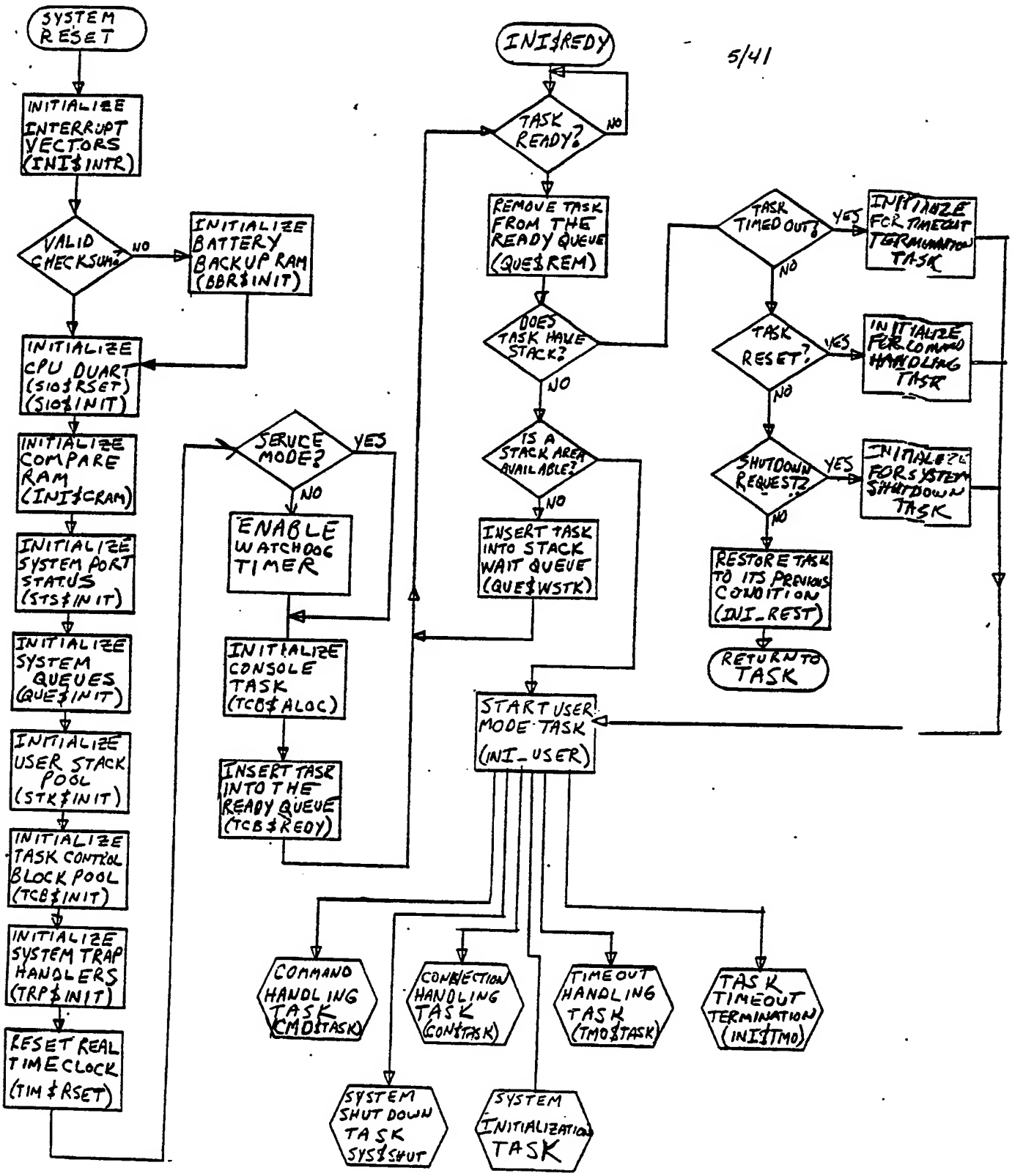

SYSTEM RESET
INITIALIZE INTERRUPT VECTORS (INI$INTR)
VALID CHECKSUM?
NO
INITIALIZE BATTERY BACKUP RAM (BBR$INIT)
INITIALIZE CPU DUART (S10$RSET) (S10$INIT)
INITIALIZE COMPARE RAM (INI$CRAM)
INITIALIZE SYSTEM PORT STATUS (STS$INIT)
INITIALIZE SYSTEM QUEUES (QUE$INIT)
INITIALIZE USER STACK POOL (STK$INIT)
INITIALIZE TASK CONTROL BLOCK POOL (TCB$INIT)
INITIALIZE SYSTEM TRAP HANDLERS (TRP$INIT)
RESET REAL TIME CLOCK (TIM$RSET)
SERVICE MODE?
YES
NO
ENABLE WATCHDOG TIMER
INITIALIZE CONSOLE TASK (TCB$ALOC)
INSERT TASK INTO THE READY QUEUE (TCB$READY)
INI$READY
TASK READY?
NO
REMOVE TASK FROM THE READY QUEUE (QUE$REM)
DOES TASK HAVE STACK?
NO
IS A STACK AREA AVAILABLE?
NO
INSERT TASK INTO STACK WAIT QUEUE (QUE$WSTK)
TASK TIMED OUT?
YES
INITIALIZE FOR TIMEOUT TERMINATION TASK
NO
TASK RESET?
YES
INITIALIZE FOR COMMAND HANDLING TASK
NO
SHUTDOWN REQUEST?
YES
INITIALIZE FOR SYSTEM SHUTDOWN TASK
NO
RESTORE TASK TO ITS PREVIOUS CONDITION (INI_REST)
RETURN TO TASK
START USER MODE TASK (INI_USER)
COMMAND HANDLING TASK (CMD$TASK)
CONNECTION HANDLING TASK (CON$TASK)
TIMEOUT HANDLING TASK (TMO$TASK)
TASK TIMEOUT TERMINATION (INI$TMO)
SYSTEM SHUT DOWN TASK SYS$SHUT
SYSTEM INITIALIZATION TASK

FIG. 5A

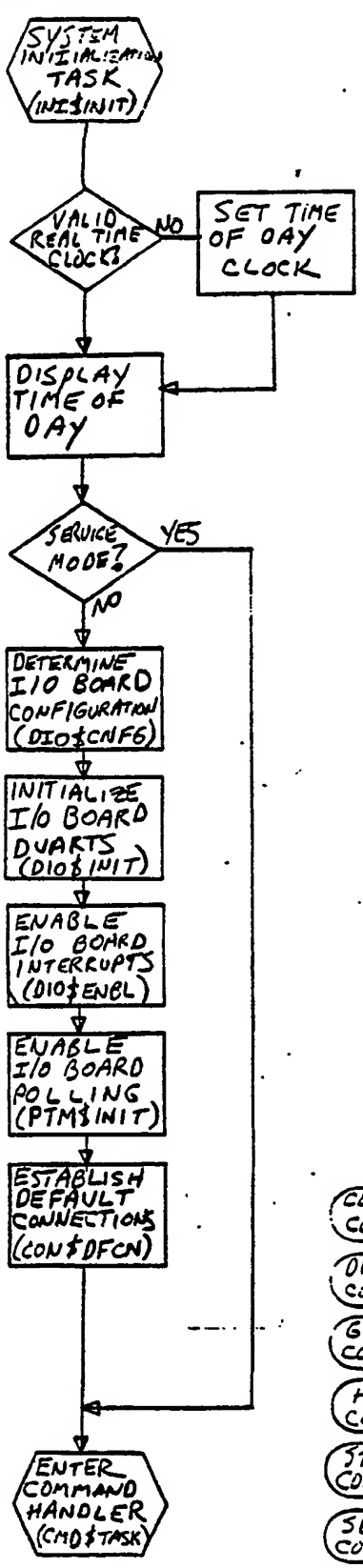

SYSTEM INITIALIZATION TASK (INI$INIT)
VALID REAL TIME CLOCK?
NO
SET TIME OF DAY CLOCK
DISPLAY TIME OF DAY
SERVICE MODE?
YES
NO
DETERMINE I/O BOARD CONFIGURATION (DIO$CNFG)
INITIALIZE I/O BOARD DUARTS (DIO$INIT)
ENABLE I/O BOARD INTERRUPTS (DIO$ENBL)
ENABLE I/O BOARD POLLING (PTM$INIT)
ESTABLISH DEFAULT CONNECTIONS (CON$DFCN)
ENTER COMMAND HANDLER (CMD$TASK)

FIG 5B

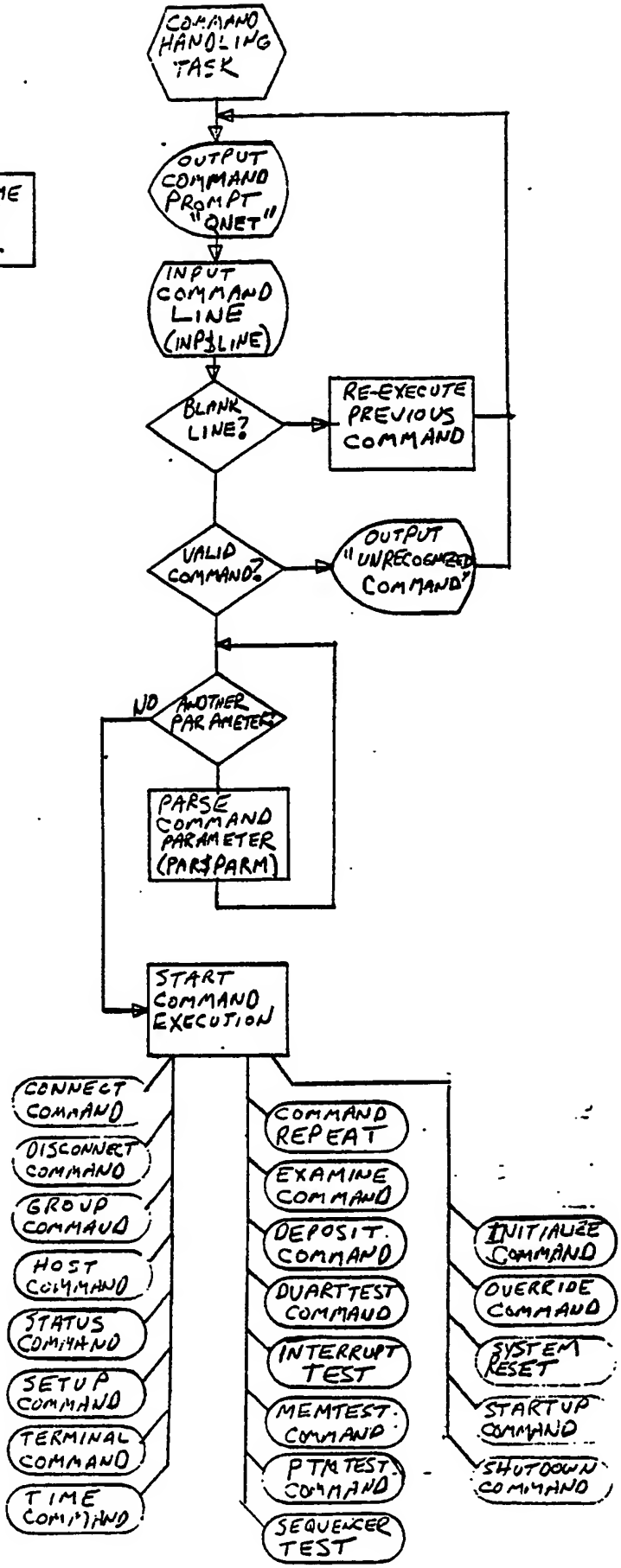

COMMAND HANDLING TASK
OUTPUT COMMAND PROMPT "QNET"
INPUT COMMAND LINE (INP$LINE)
BLANK LINE?
RE-EXECUTE PREVIOUS COMMAND
VALID COMMAND?
OUTPUT "UNRECOGNIZED COMMAND"
NO
ANOTHER PARAMETER?
PARSE COMMAND PARAMETER (PAR$PARM)
START COMMAND EXECUTION
CONNECT COMMAND
DISCONNECT COMMAND
GROUP COMMAND
HOST COMMAND
STATUS COMMAND
SETUP COMMAND
TERMINAL COMMAND
TIME COMMAND
COMMAND REPEAT
EXAMINE COMMAND
DEPOSIT COMMAND
DUARTTEST COMMAND
INTERRUPT TEST
MEMTEST COMMAND
PTM TEST COMMAND
SEQUENCER TEST
INITIALIZE COMMAND
OVERRIDE COMMAND
SYSTEM RESET
STARTUP COMMAND
SHUTDOWN COMMAND

FIG. 5C

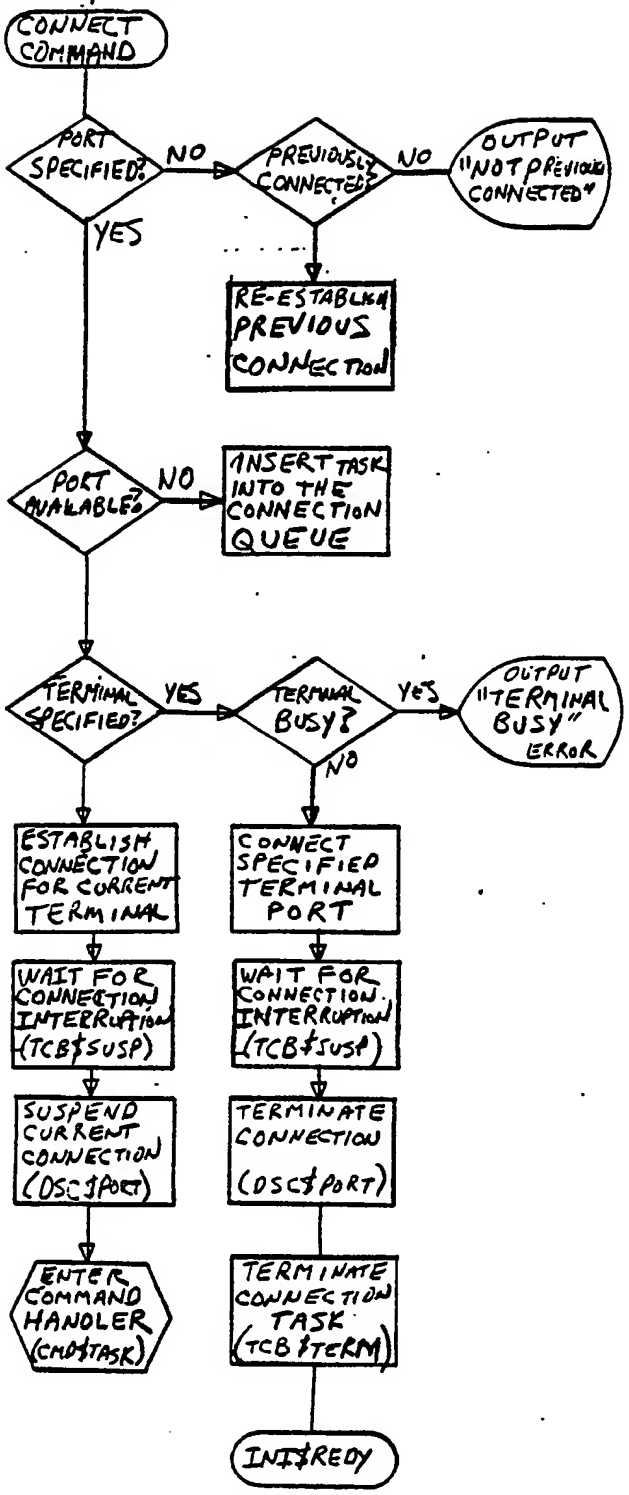

CONNECT COMMAND
PORT SPECIFIED?
NO
PREVIOUSLY CONNECTED?
NO
OUTPUT "NOT PREVIOUSLY CONNECTED"
YES
RE-ESTABLISH PREVIOUS CONNECTION
PORT AVAILABLE?
NO
INSERT TASK INTO THE CONNECTION QUEUE
TERMINAL SPECIFIED?
YES
TERMINAL BUSY?
YES
OUTPUT "TERMINAL BUSY" ERROR
NO
ESTABLISH CONNECTION FOR CURRENT TERMINAL
CONNECT SPECIFIED TERMINAL PORT
WAIT FOR CONNECTION INTERRUPTION (TCB$SUSP)
WAIT FOR CONNECTION INTERRUPTION (TCB$SUSP)
SUSPEND CURRENT CONNECTION (DSC$PORT)
TERMINATE CONNECTION (DSC$PORT)
ENTER COMMAND HANDLER (CMD$TASK)
TERMINATE CONNECTION TASK (TCB$TERM)
INI$READY

FIG. 6A.

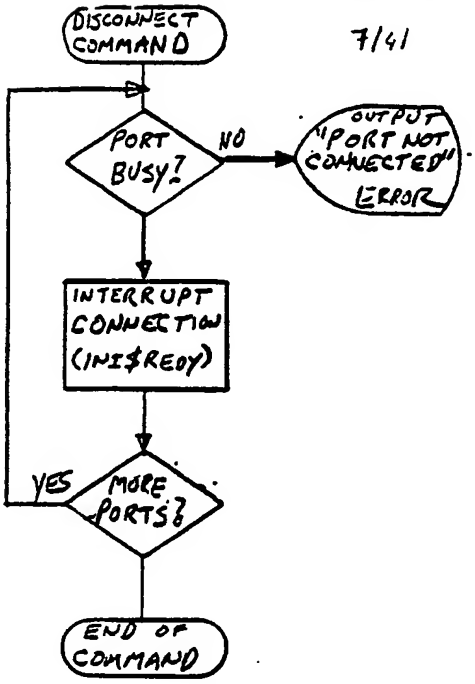

DISCONNECT COMMAND
PORT BUSY?
NO
OUTPUT "PORT NOT CONNECTED" ERROR
INTERRUPT CONNECTION (INI$READY)
YES
MORE PORTS?
END OF COMMAND

FIG. 6B

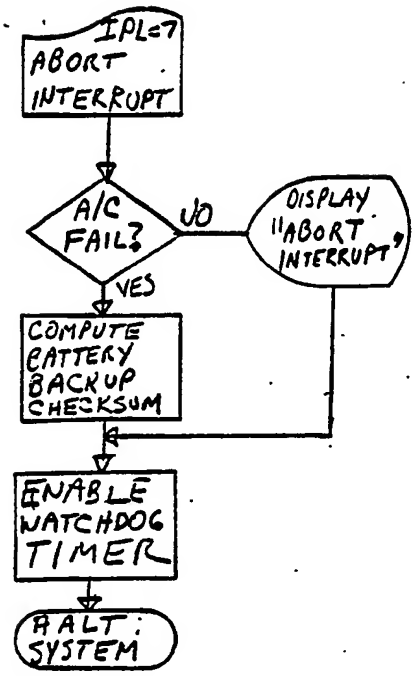


FIG. 7A

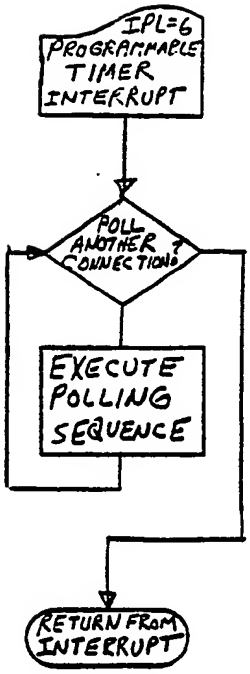


FIG. 7B

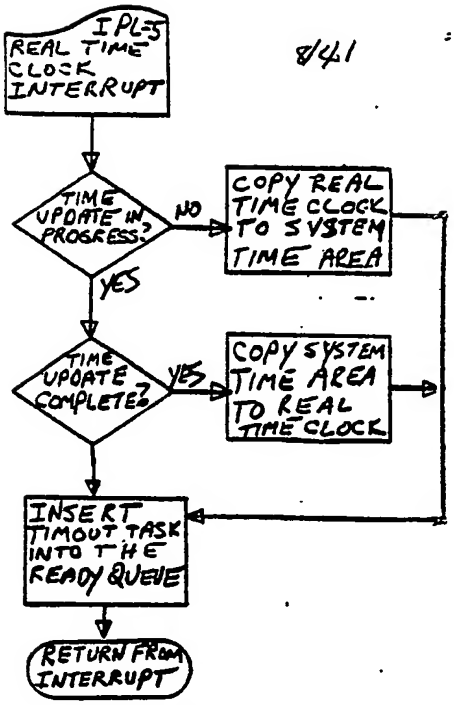


FIG 7C

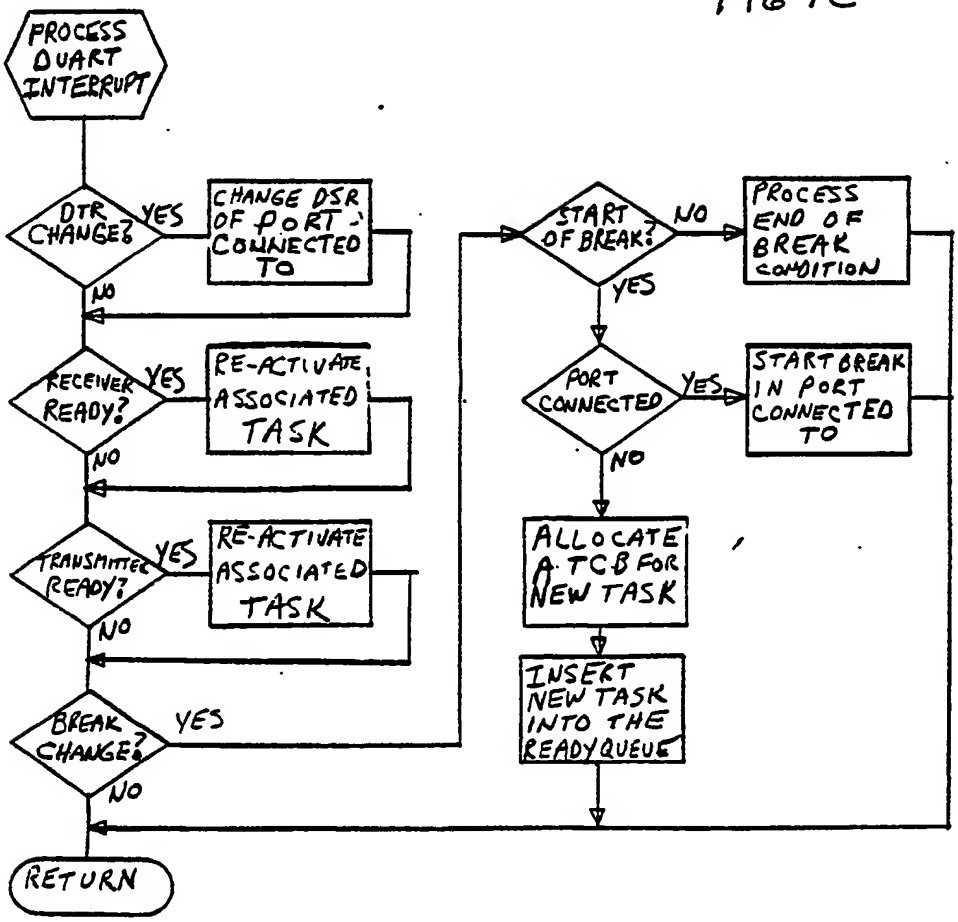


FIG 7H.

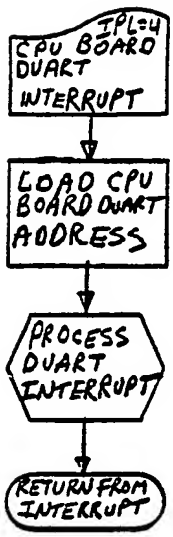


FIG. 7D

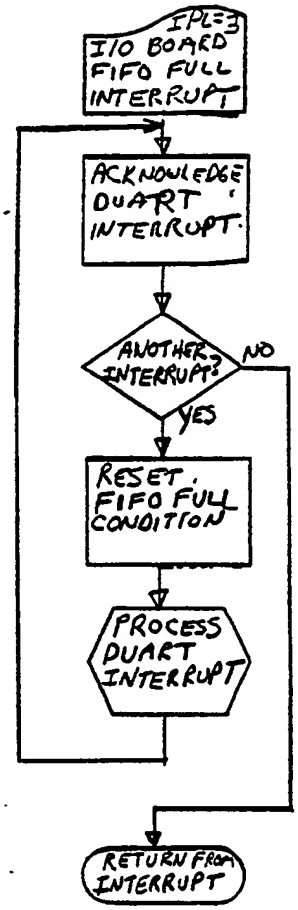


FIG. 7E

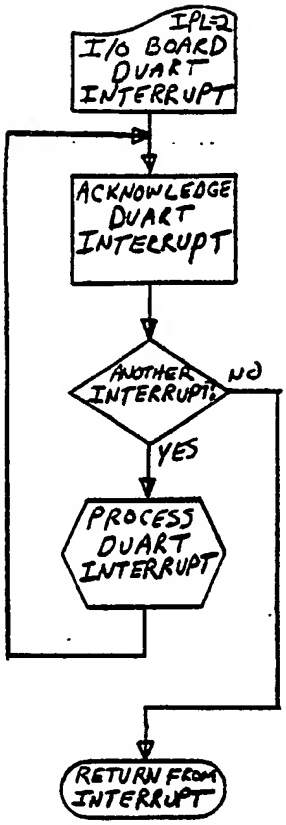


FIG. 7F.

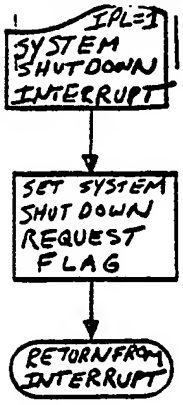


FIG. 7G

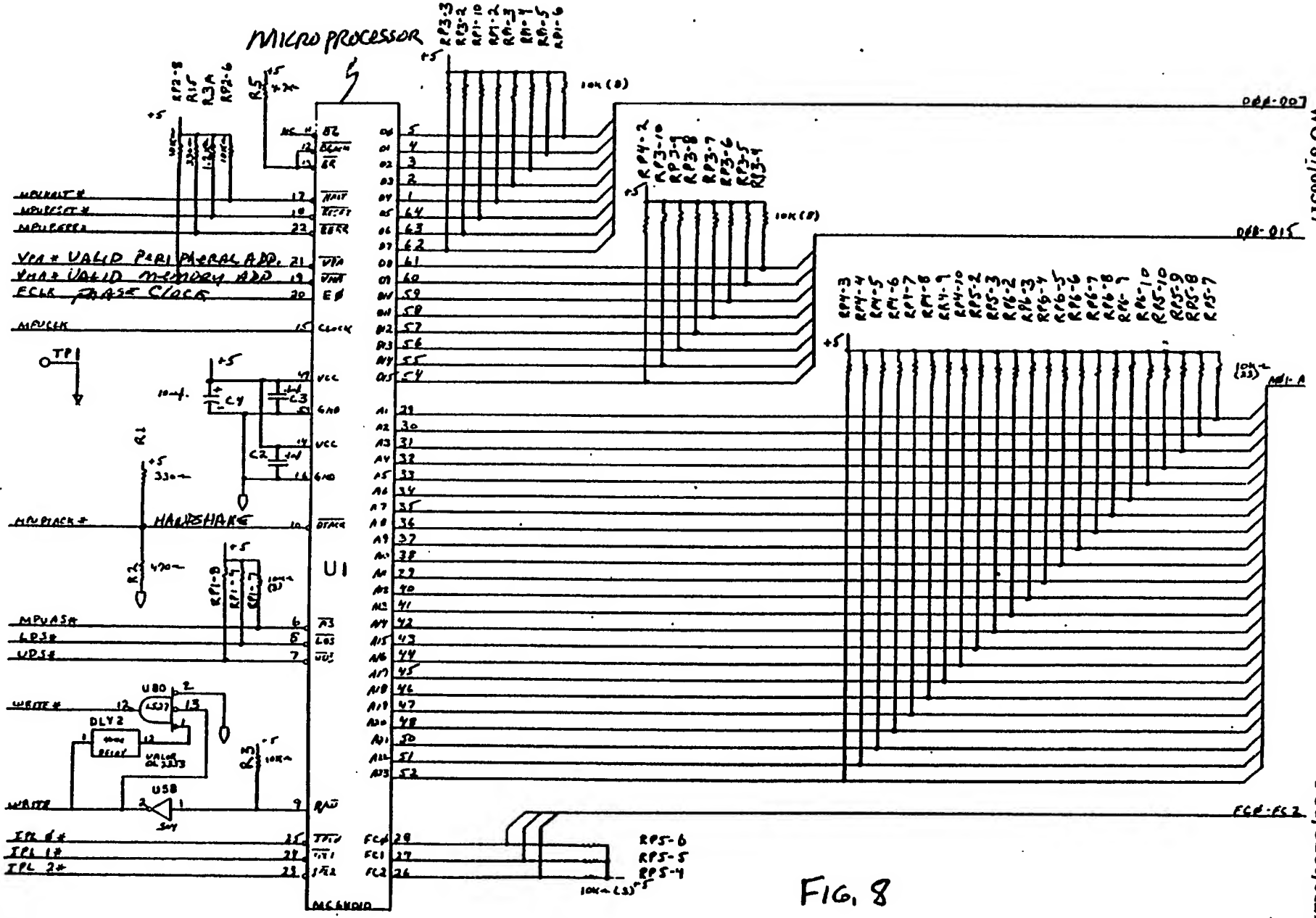


FIG. 8

* IS INTENDED TO be a NOT SYMBOL

WO 87/00317
PCT/US86/01219

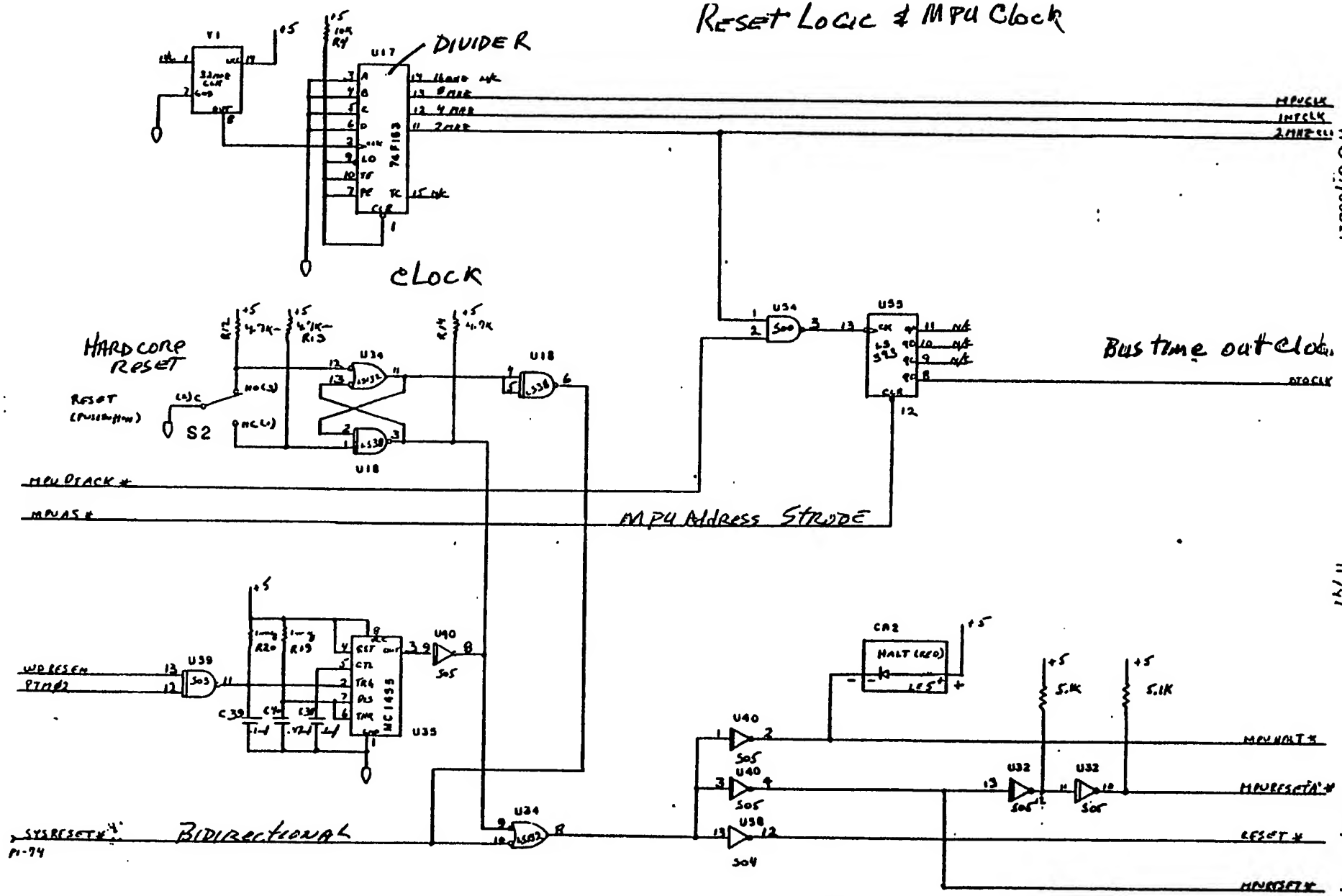

Reset Logic & MPU Clock
DIVIDER
clock
HARDCORE RESET
RESET (PUSH-BUTTON)
S2
Bus time out clock
MPU Address STROBE
BIDIRECTIONAL
MPUCLK
INTCLK
MPUBACK*
MPUAS*
MPUHALT*
RESET*
SYSRESET*
HALT (LED)
5.1K
74F163
MC1455

FIG 9

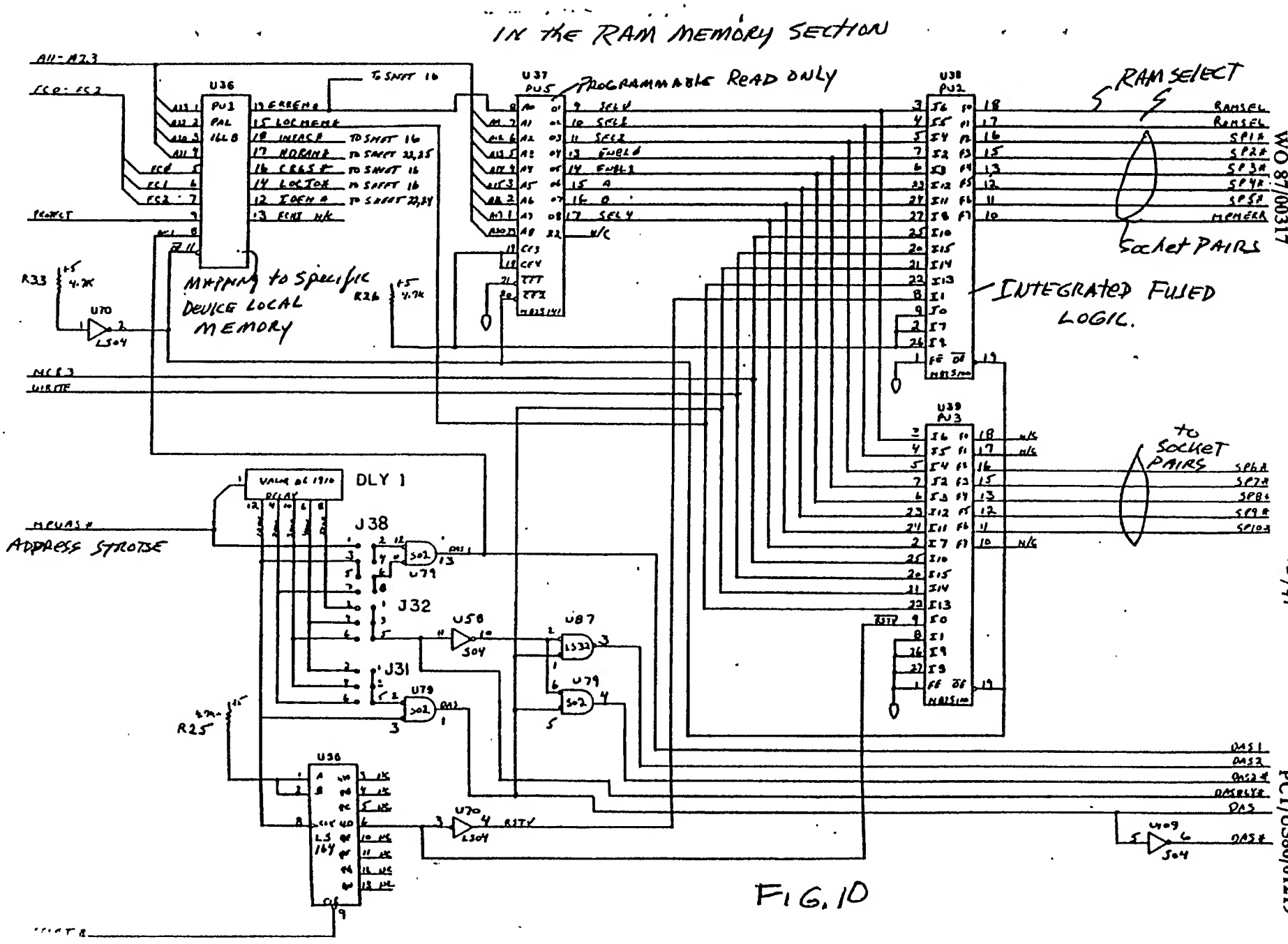

IN THE RAM MEMORY SECTION
A11-A23
FC0-FC2
U36
PU1
PAL
16L8
TO SHIFT 16
MAPPING to specific DEVICE LOCAL MEMORY
U37
PU5
PROGRAMMABLE READ ONLY
U38
PU2
RAM SELECT
Socket PAIRS
INTEGRATED FUSED LOGIC.
U39
PU3
to Socket PAIRS
R33
R26
R25
U70
LS04
DLY 1
J38
J32
J31
U58
U87
U79
U36
LS 164
ADDRESS STROBE
WRITE

FIG. 10

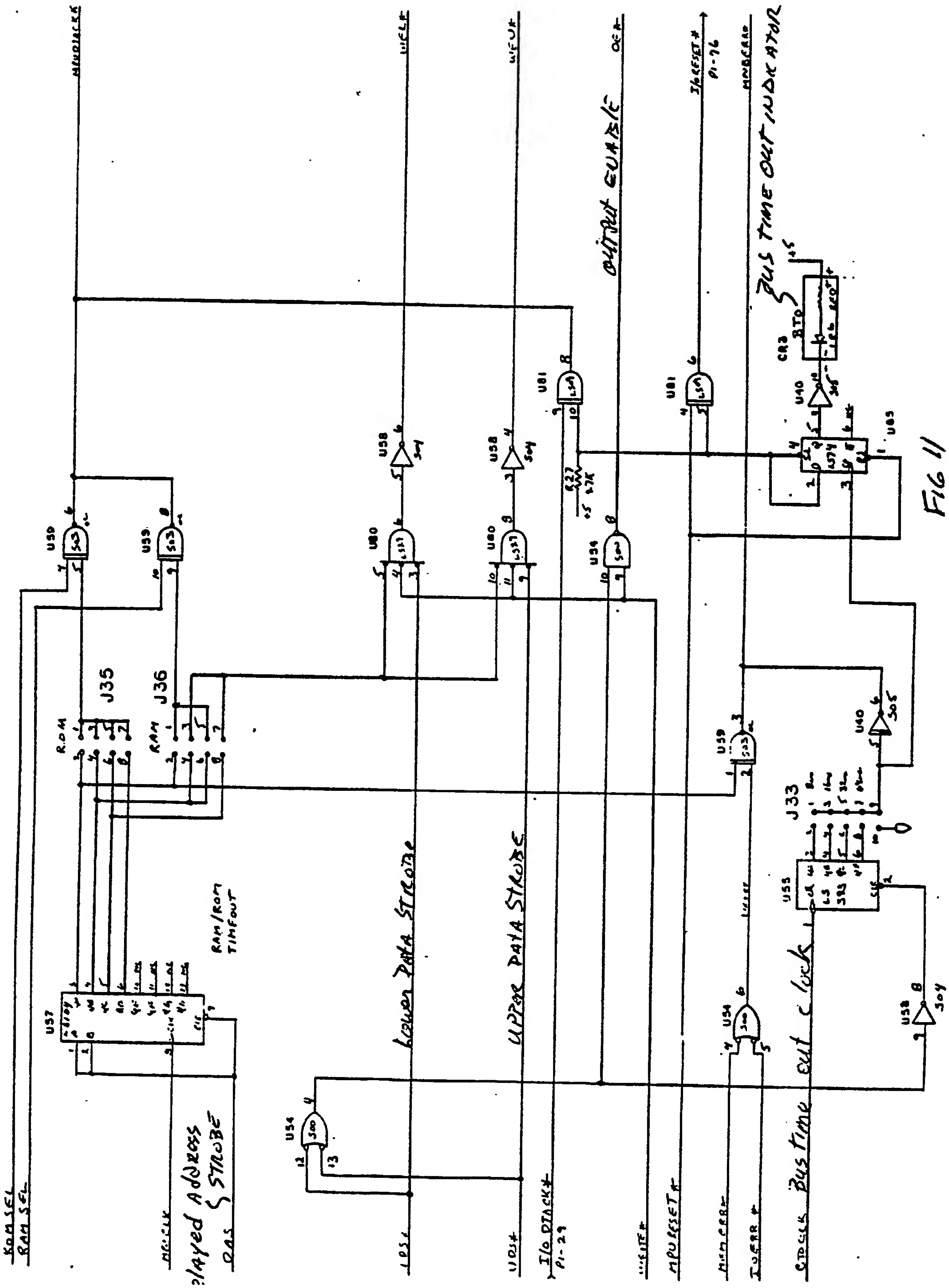

BUS TIME OUT INDICATOR
OUTPUT ENABLE
LOWER DATA STROBE
UPPER DATA STROBE
Bus time out clock
RAM/ROM TIMEOUT
J35
J36
J33
ROM
RAM
ROMSEL
RAMSEL
CR3
BTO
U37
U55

FIG 11

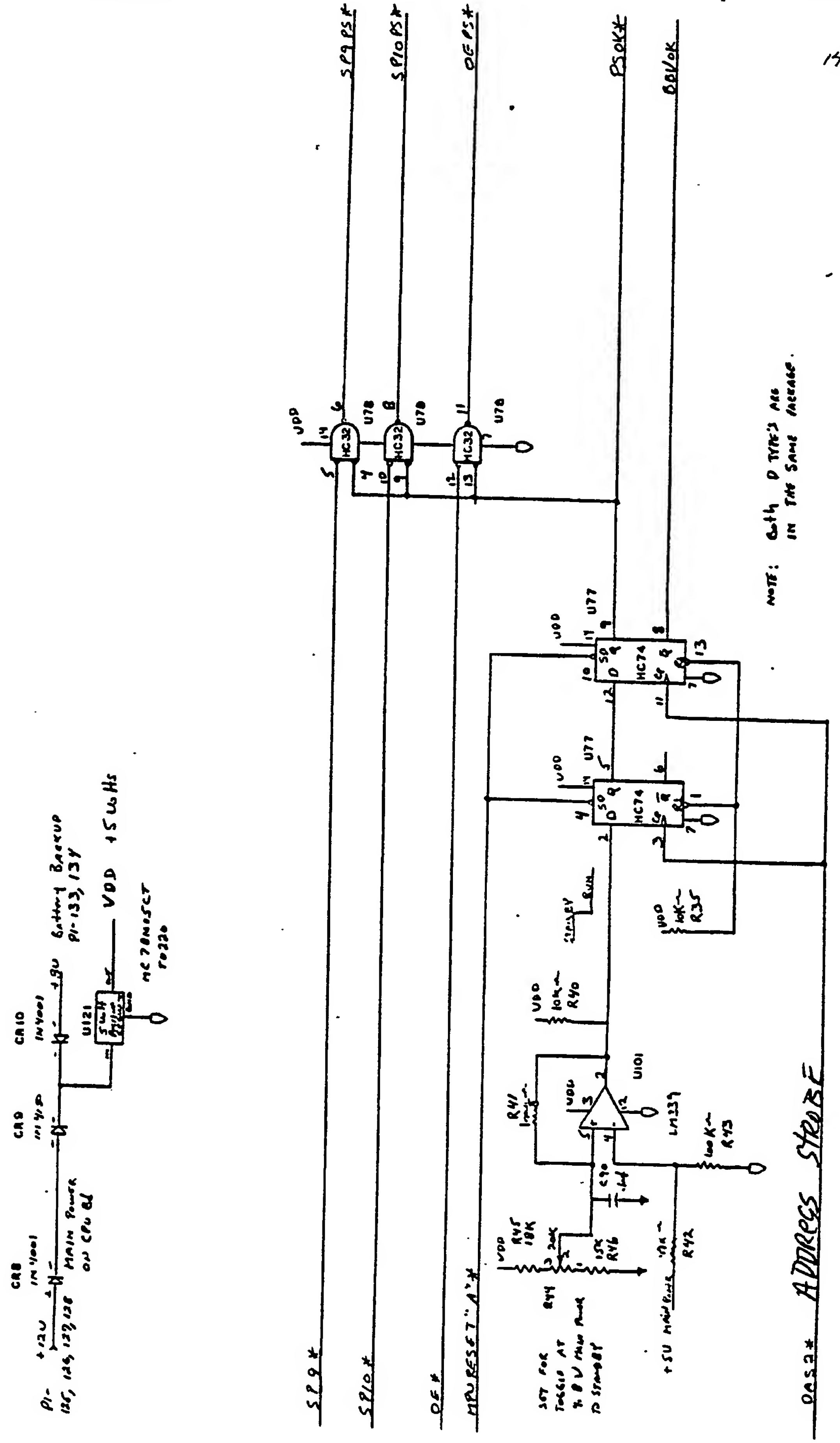

SP9PS*
SP10PS*
OEPS*
PSOK*
BBVOK
HC32
U78
HC74
U77
VDD
NOTE: Both D TYPE'S ARE IN THE SAME PACKAGE.
LM339
U101
U121
MC78M05CT
VDD +5 Volts
Battery Backup
Main Power on CPU Bd
SP9*
SP10*
OE*
MPURESET "A"*
OAS2*
ADDRESS STROBE

FIG. 12

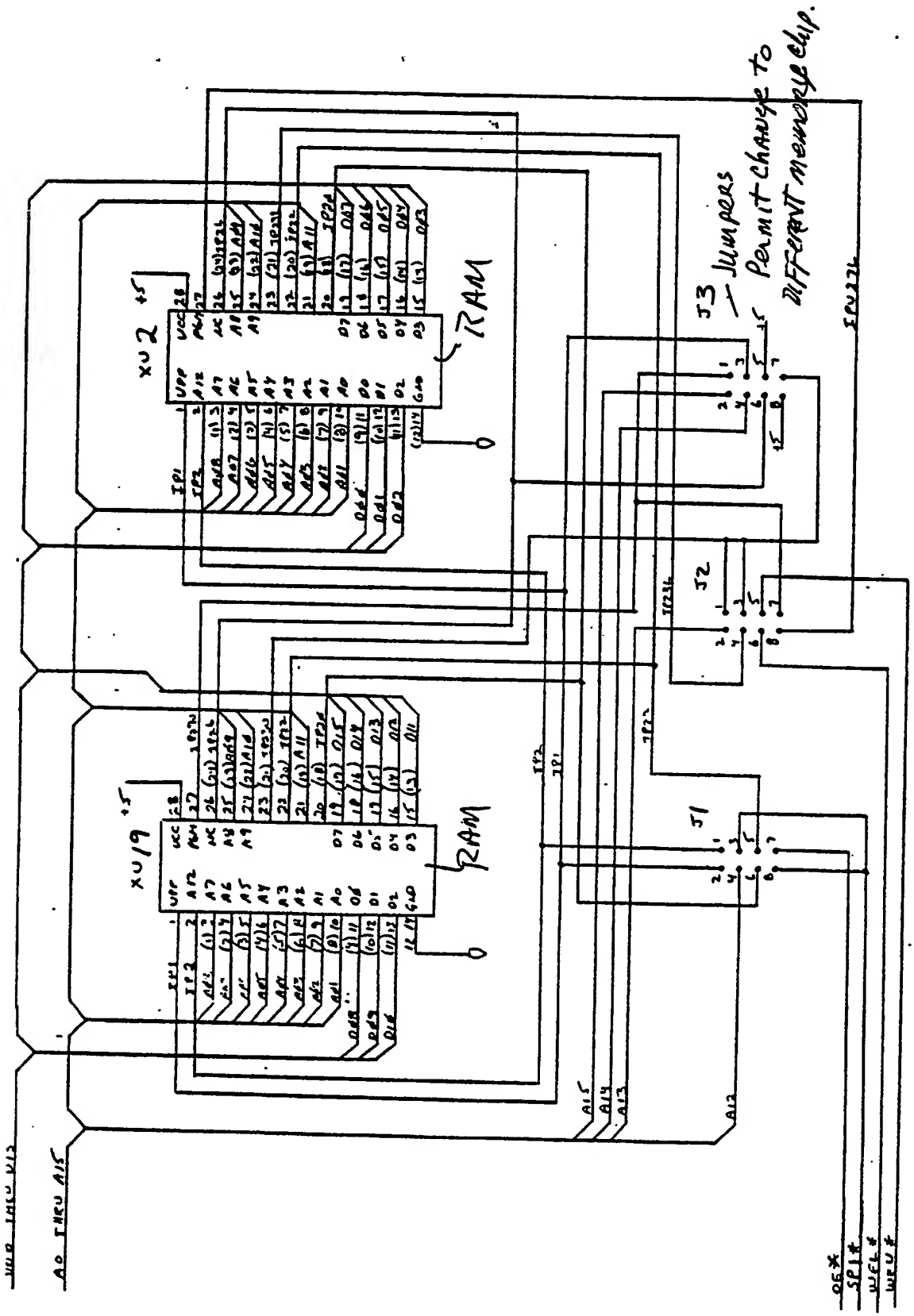


FIG. 13
(There are memory circuits corresponding to this figure with corresponding socket pins)

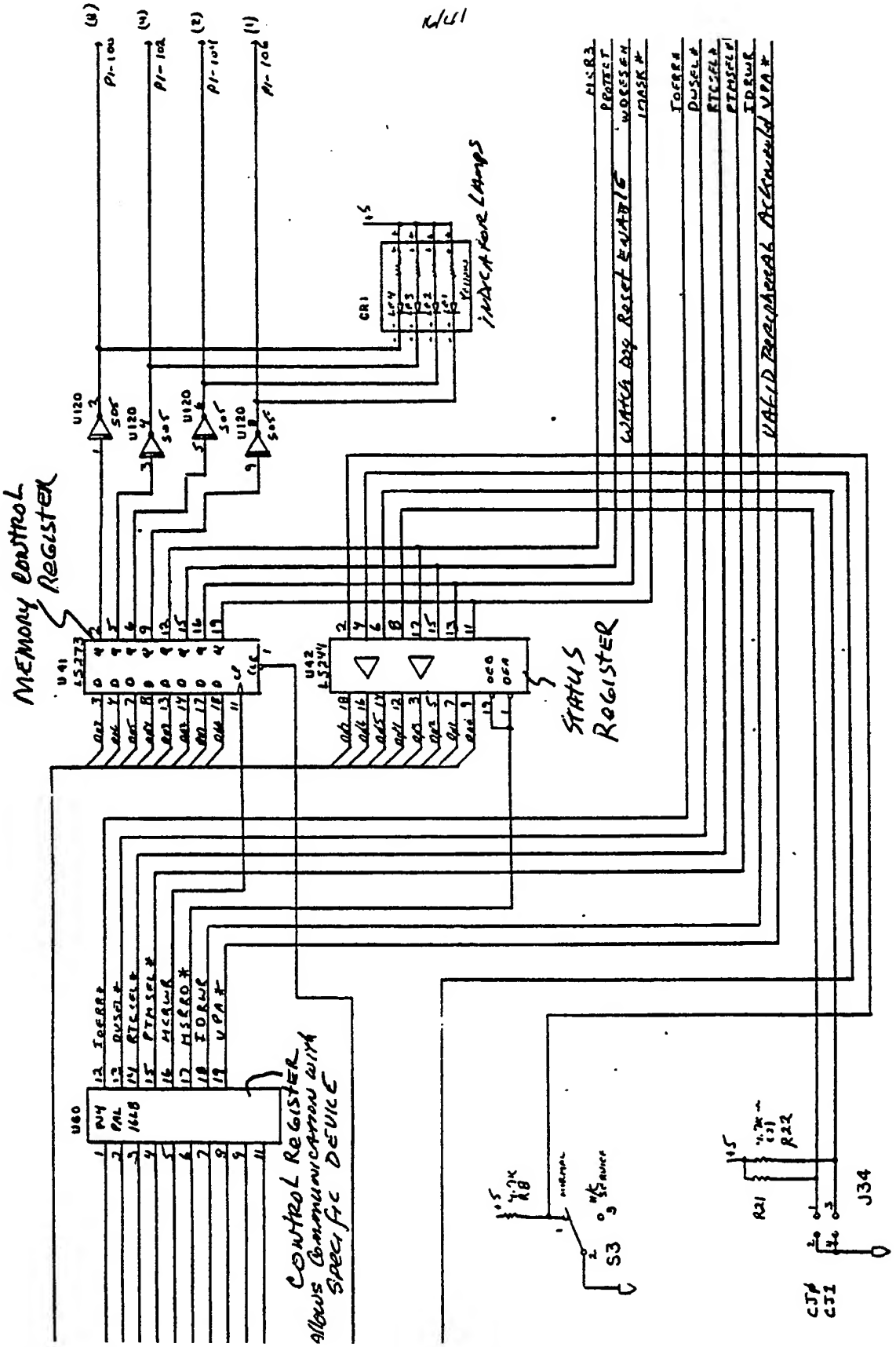

MEMORY CONTROL REGISTER
STATUS REGISTER
CONTROL REGISTER
ALLOWS COMMUNICATION WITH SPECIFIC DEVICE
INDICATOR LAMPS
WATCH DOG RESET ENABLE
VALID READHEAD ACKNOWLEDGE
U41
U42
U60
U120
CR1
S3
J34
R21
R22

FIG. 14

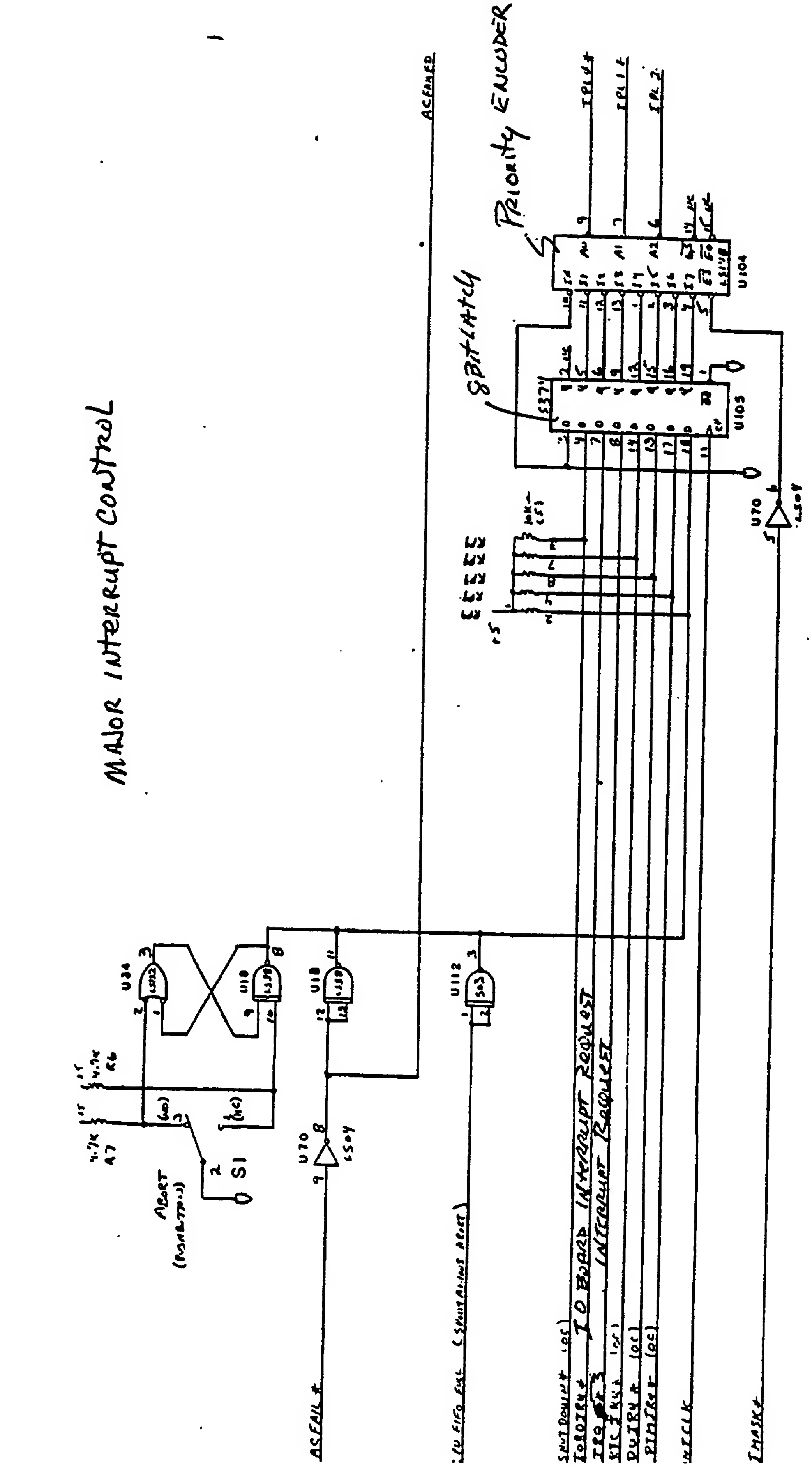

MAJOR INTERRUPT CONTROL
PRIORITY ENCODER
8BIT LATCH
ACFAILED
IPL0+
IPL1+
IPL2
U104
U105
U70
U34
U18
U112
S1
ABORT
ACFAIL+
IO BOARD INTERRUPT REQUEST
INTERRUPT REQUEST
IMASK+

FIG. 15

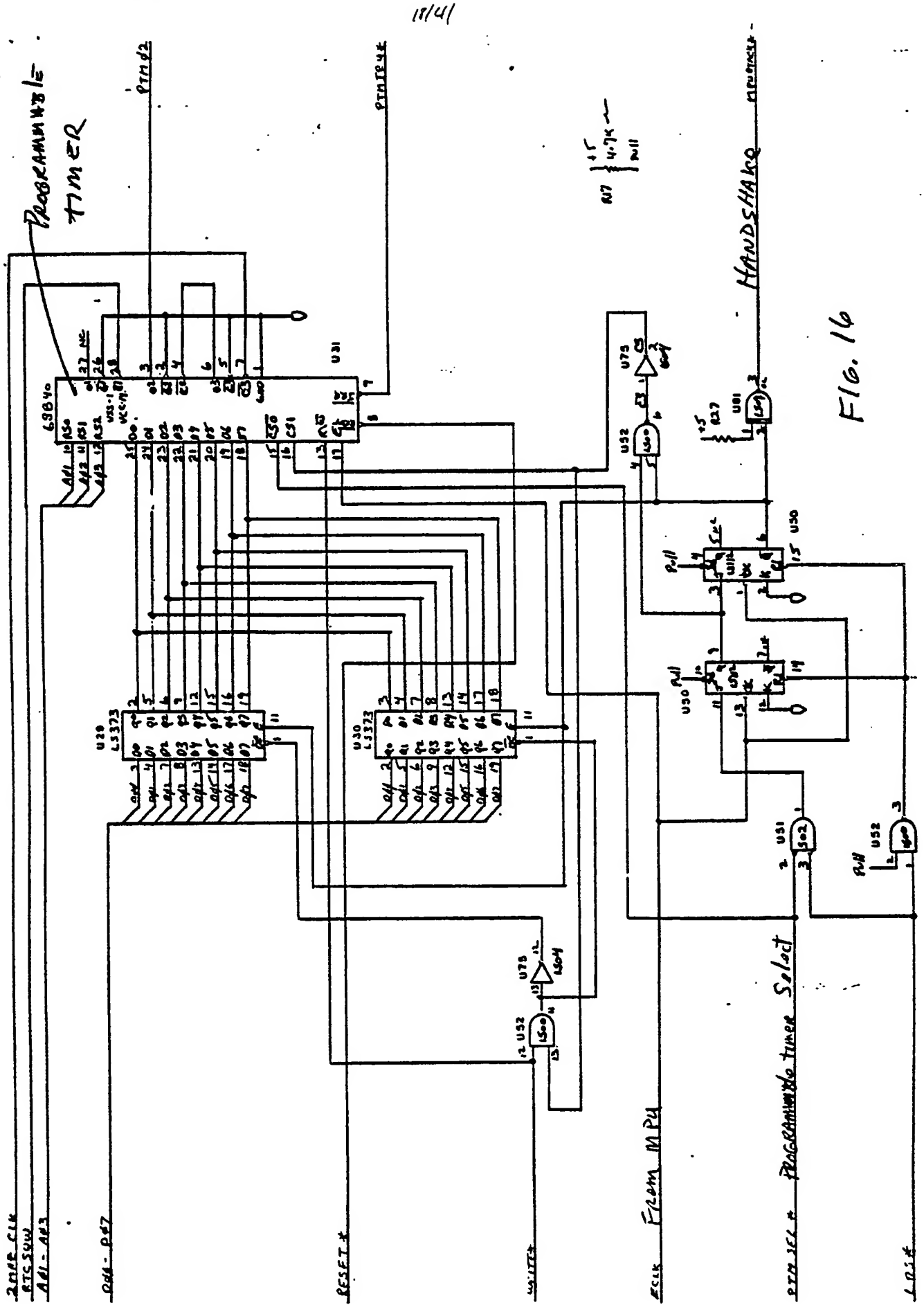
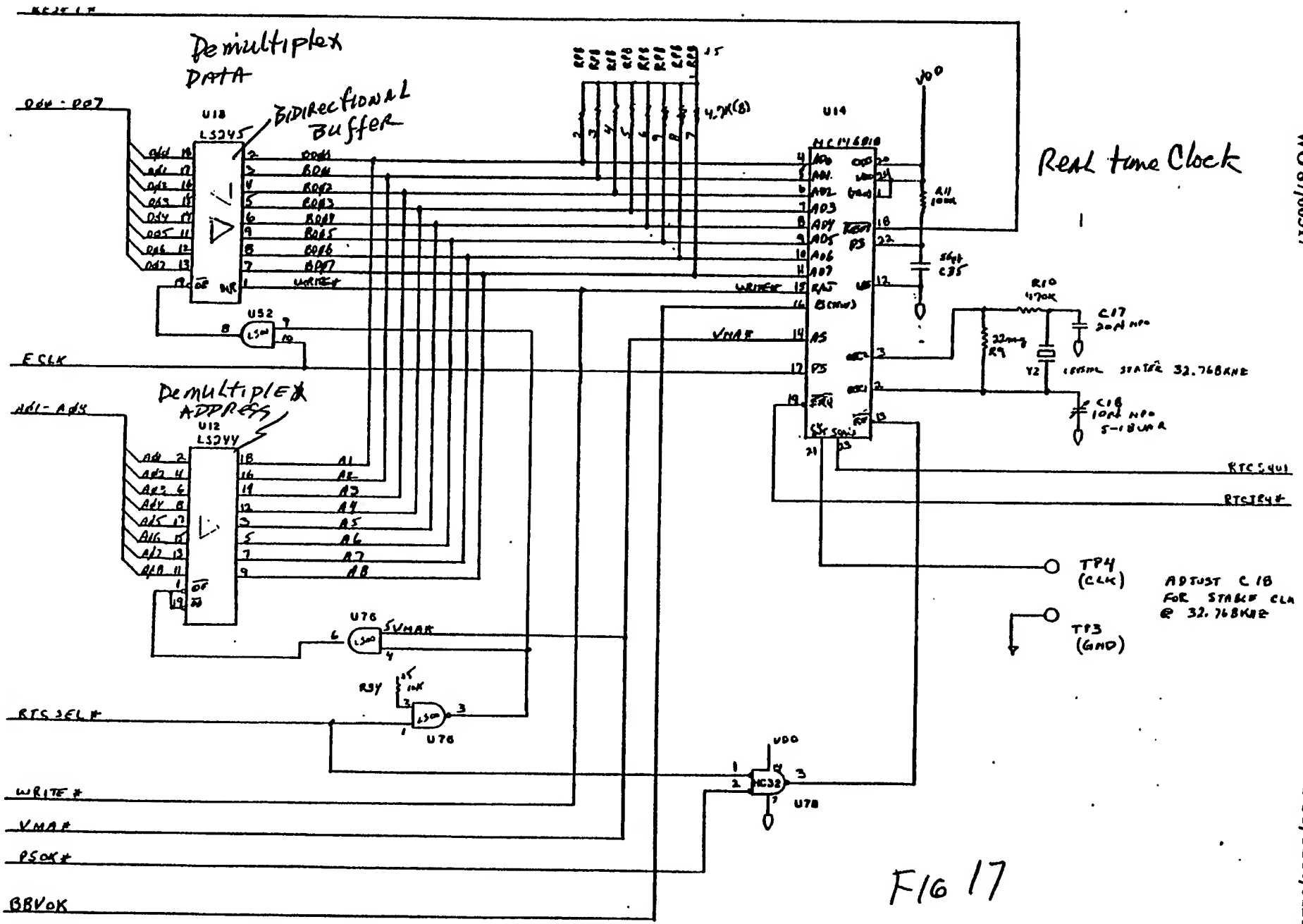
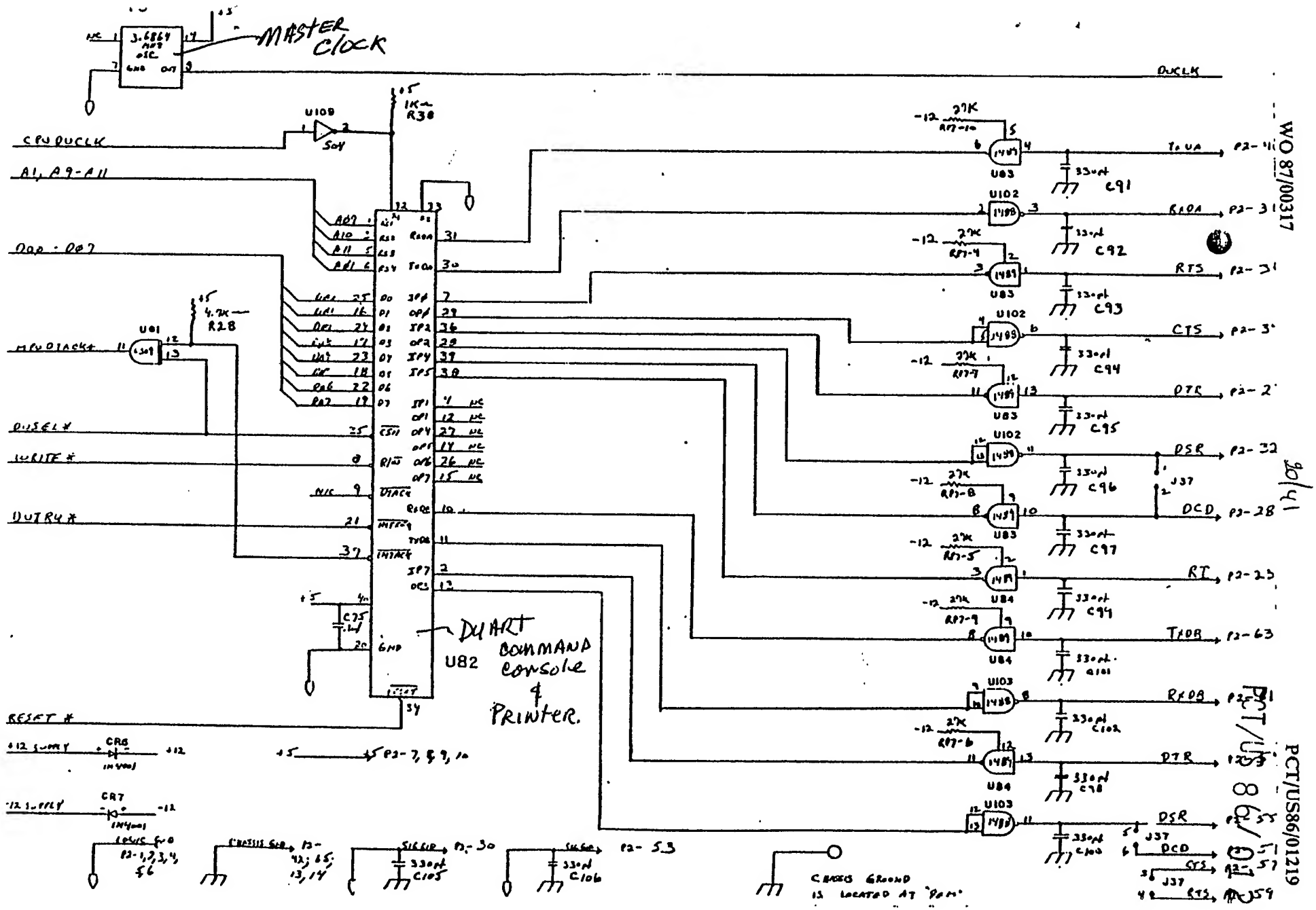

PROGRAMMABLE TIMER
HANDSHAKE
RESET#
R/W
Select
PROGRAMMABLE TIMER
FROM MPU

FIG. 16



WO 87/00317
PCT/US86/01219



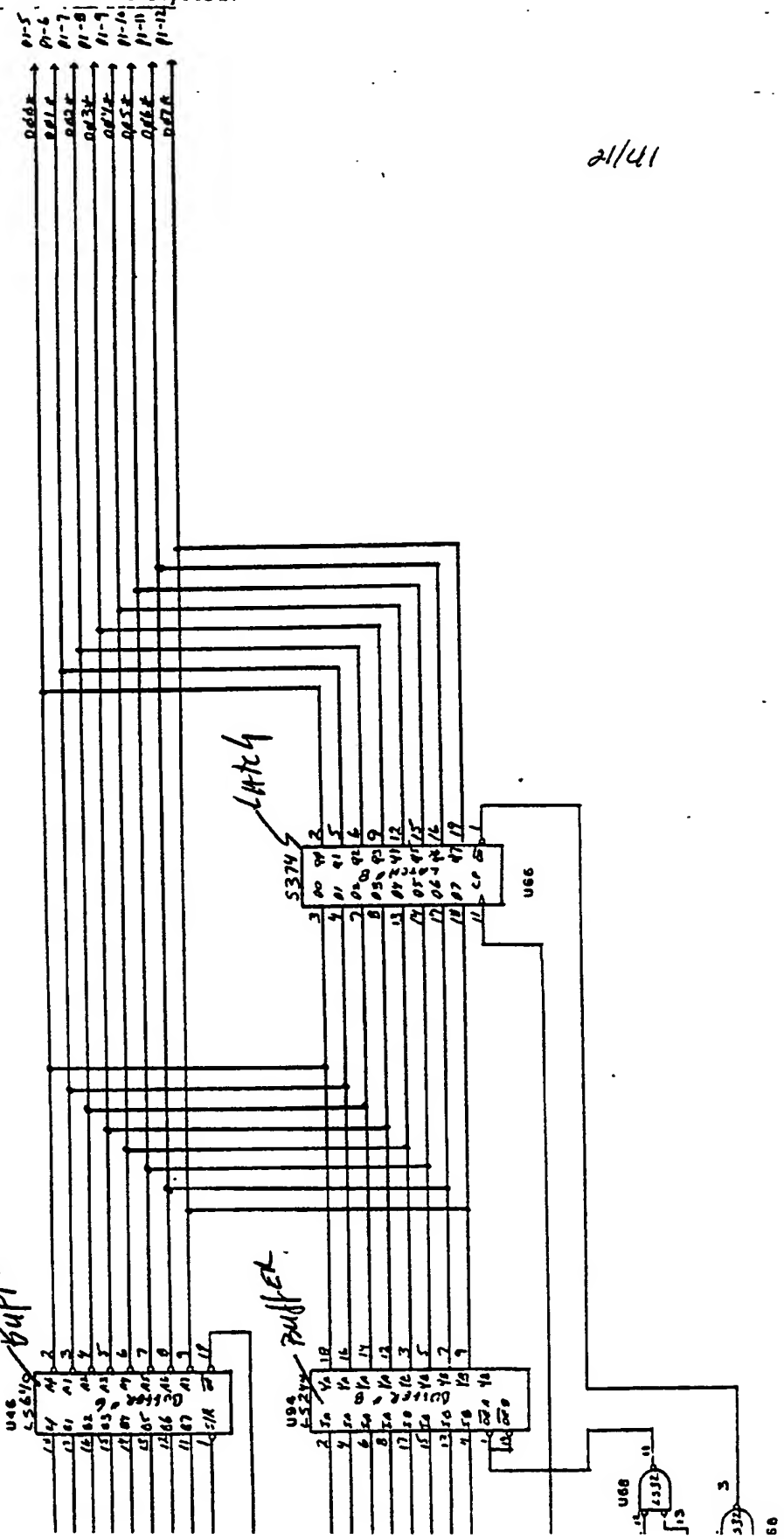


Fig 19

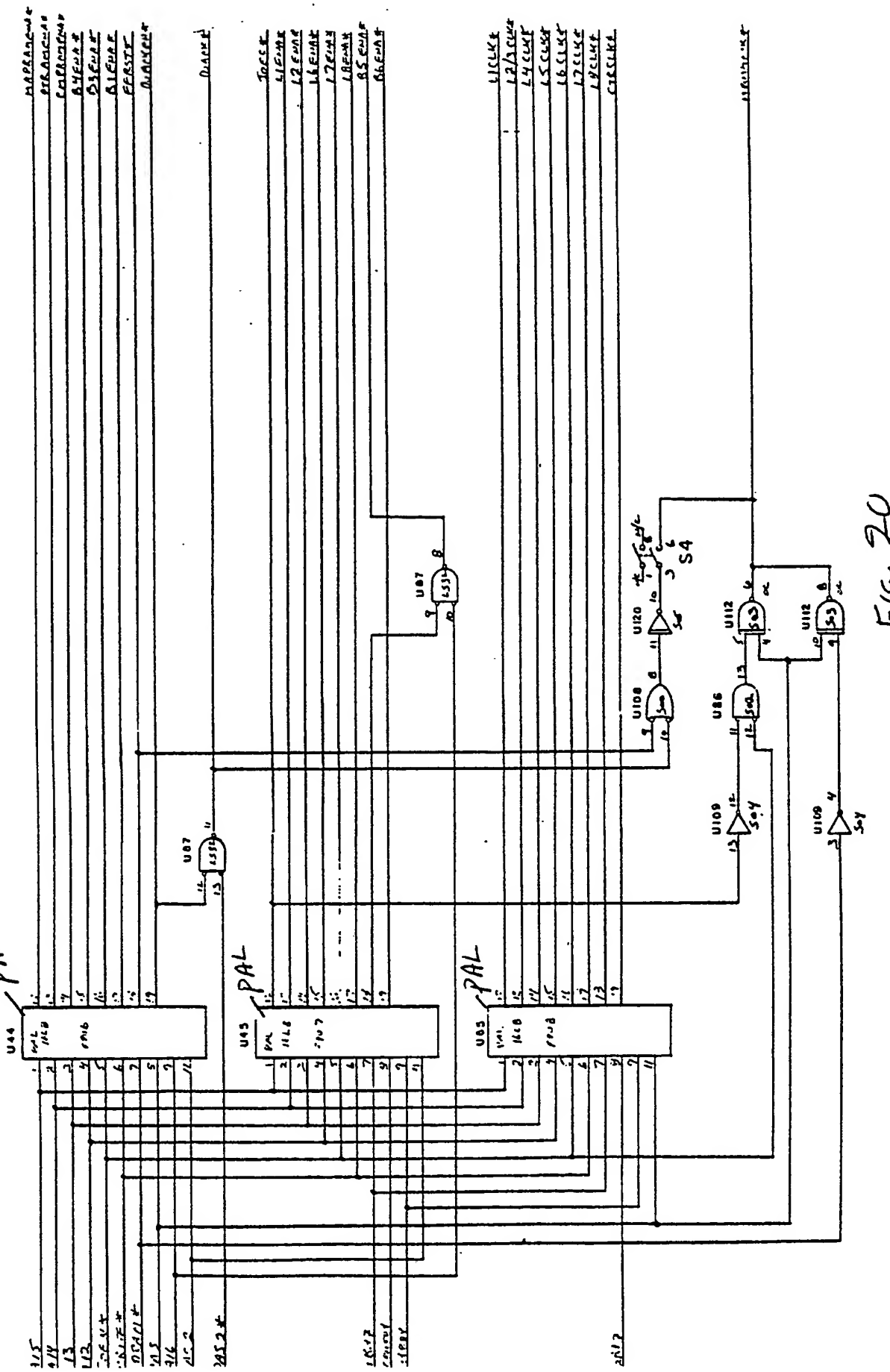


FIG. 20

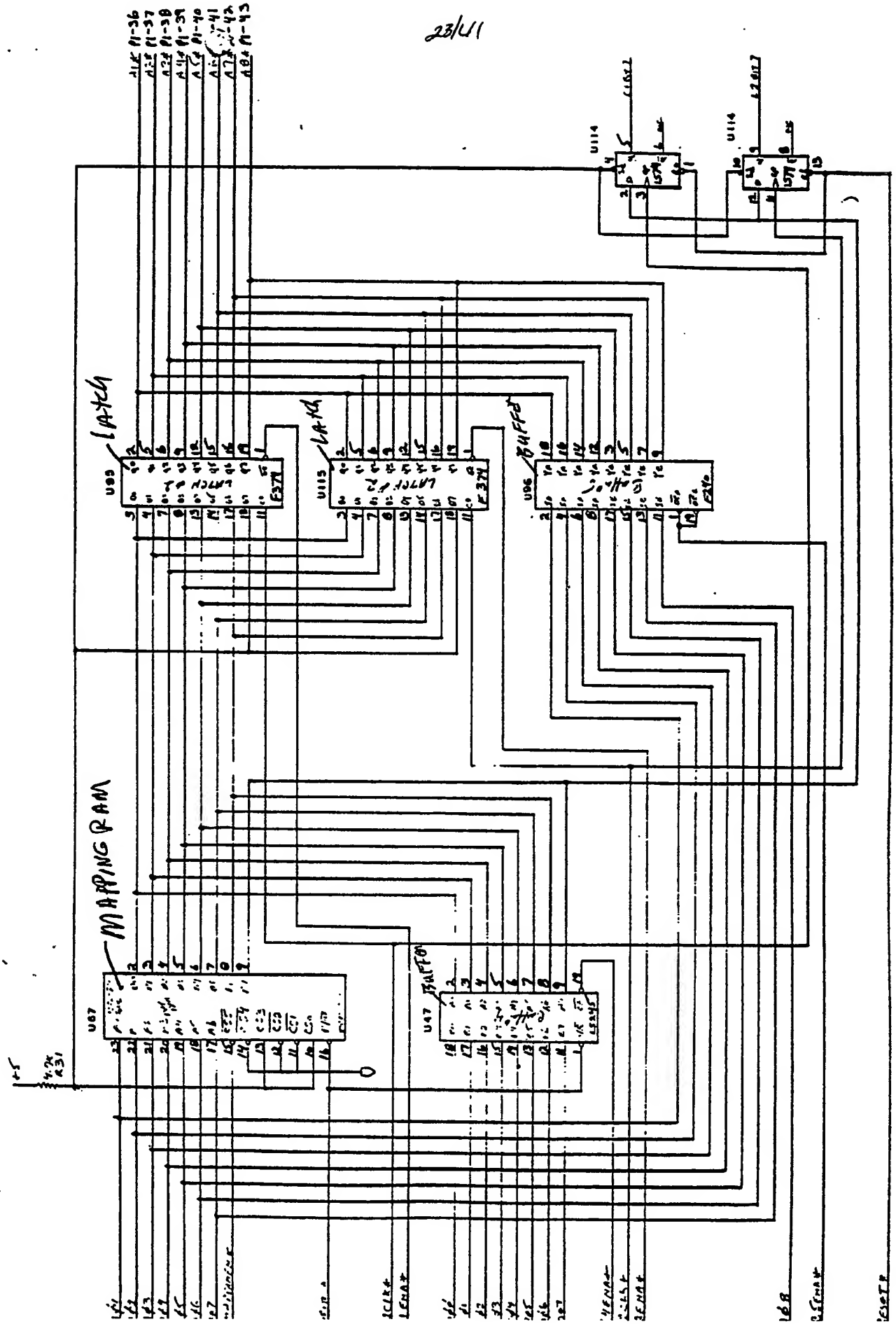


FIG 21

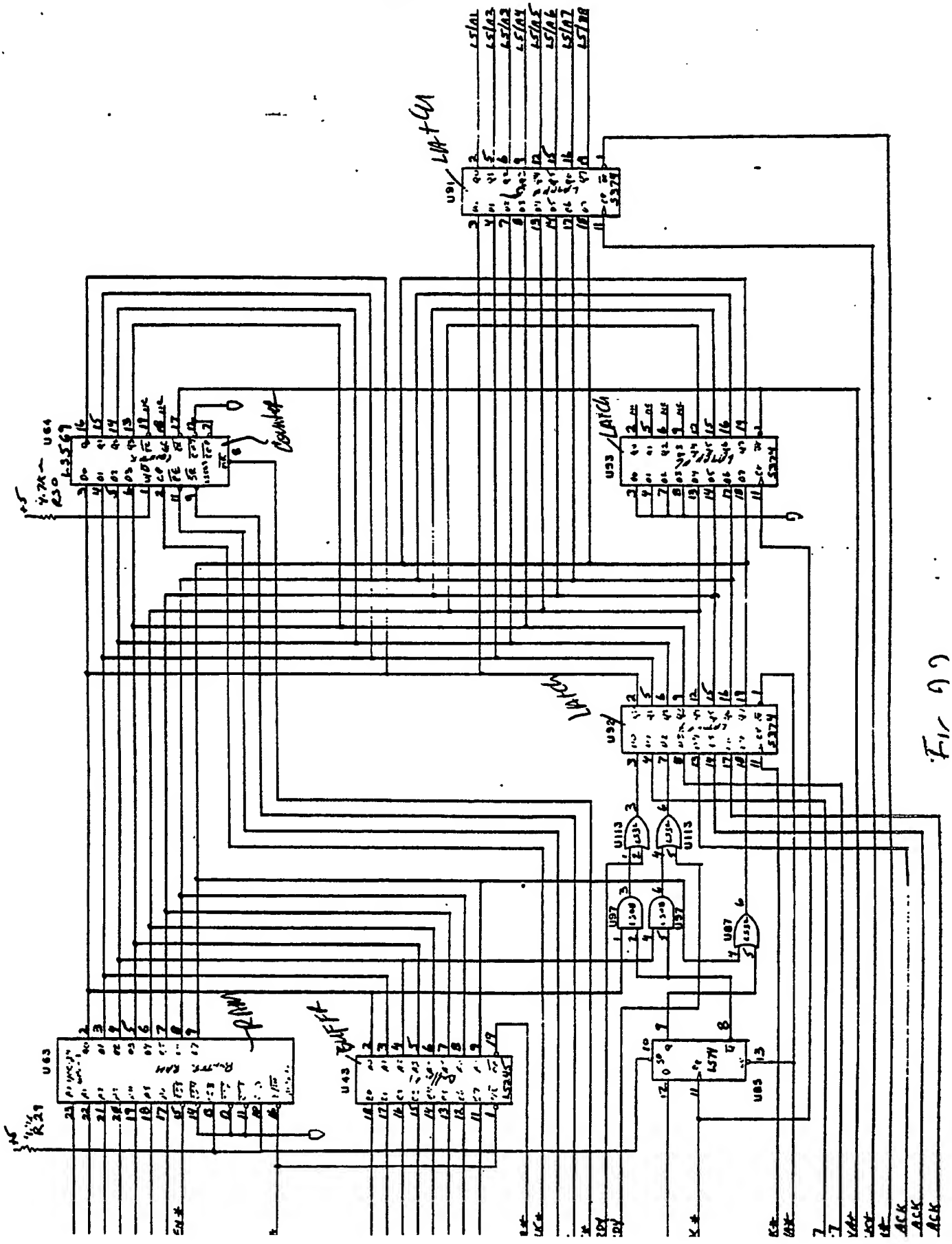


Fig 10

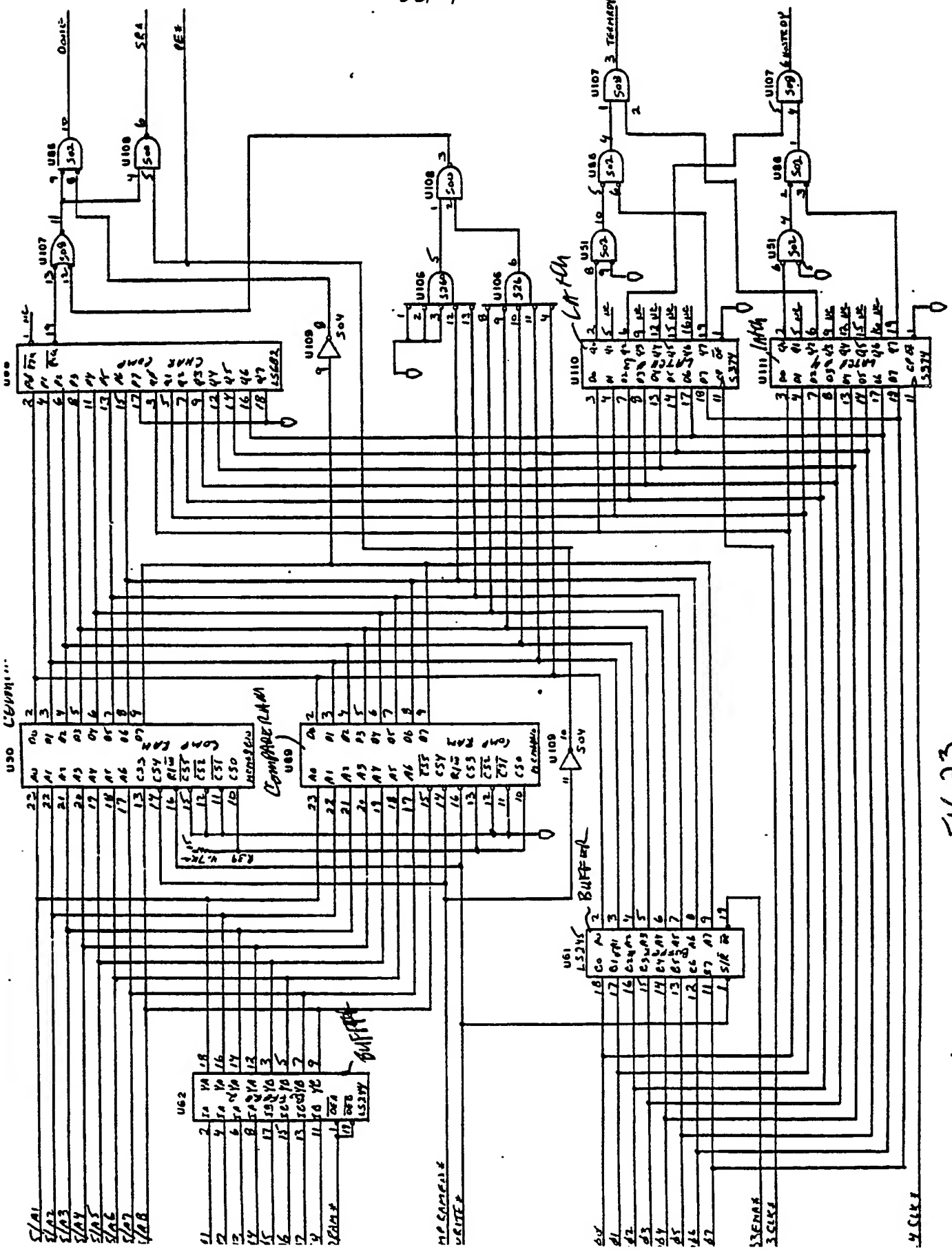


FIG 23

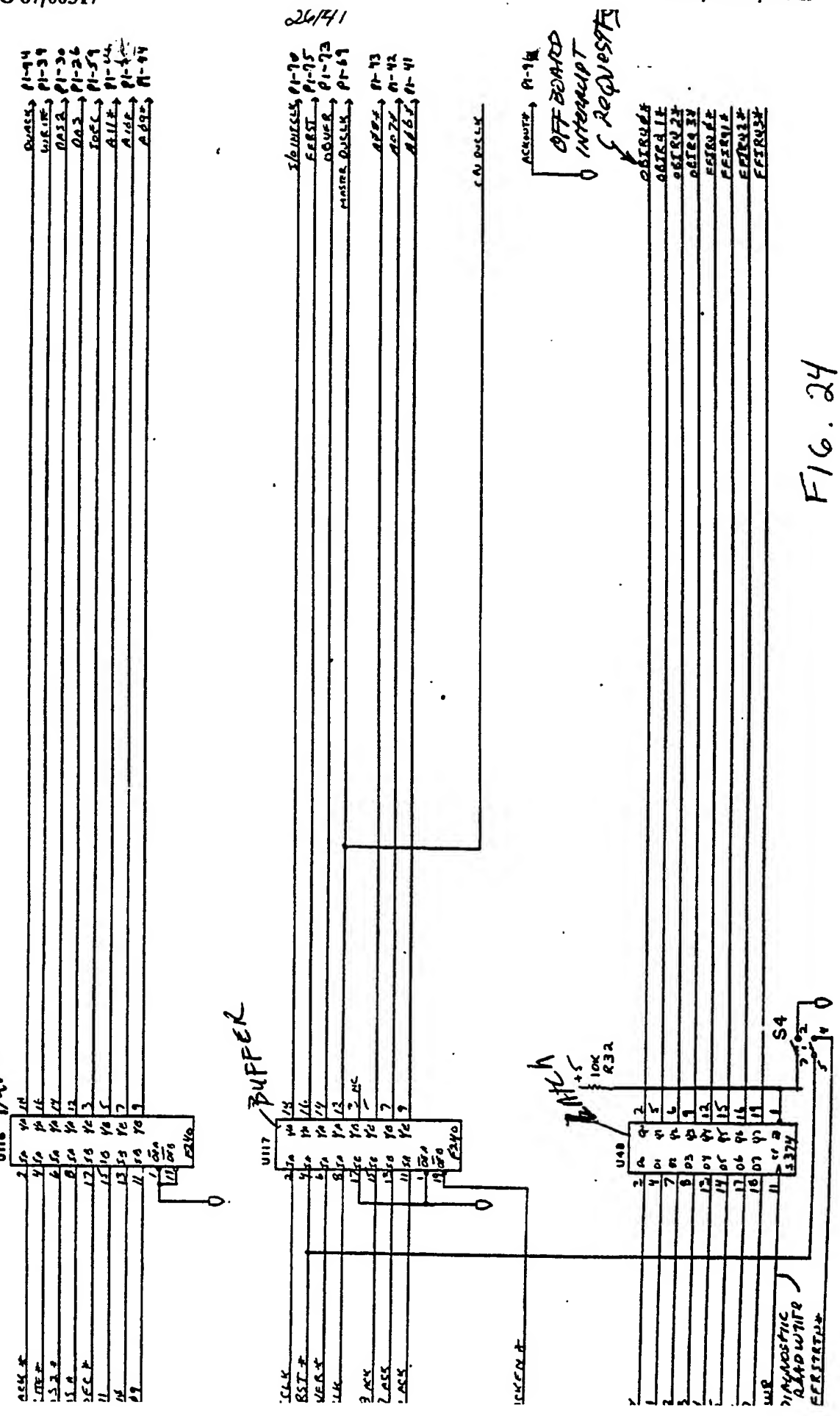

BUFFER
BUFFER
OFF BOARD INTERRUPT REQUEST
FOR SINGLE BOARD CPU ONLY DIAG USE ONLY
S4

FIG. 24

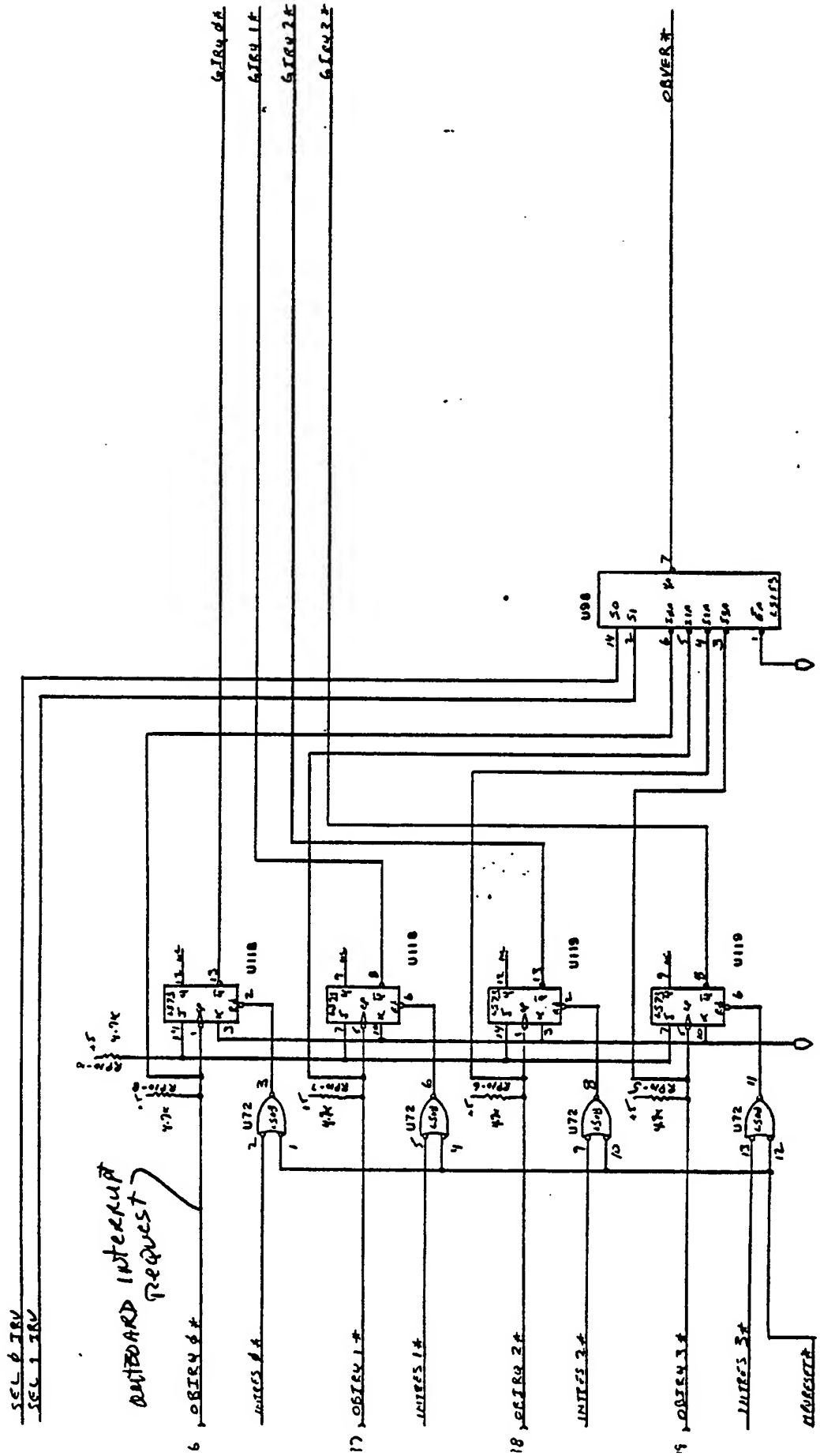

SEL 0 IRV
SEL 1 IRV
KEYBOARD INTERRUPT REQUEST
OBERR 0#
INTERR 0#
OBERR 1#
INTERR 1#
OBERR 2#
INTERR 2#
OBERR 3#
INTERR 3#
REQUEST#
6IRQ 0#
6IRQ 1#
6IRQ 2#
6IRQ 3#
OBVER#
U98
U118
U119
U72
4.7K

FIG. 25

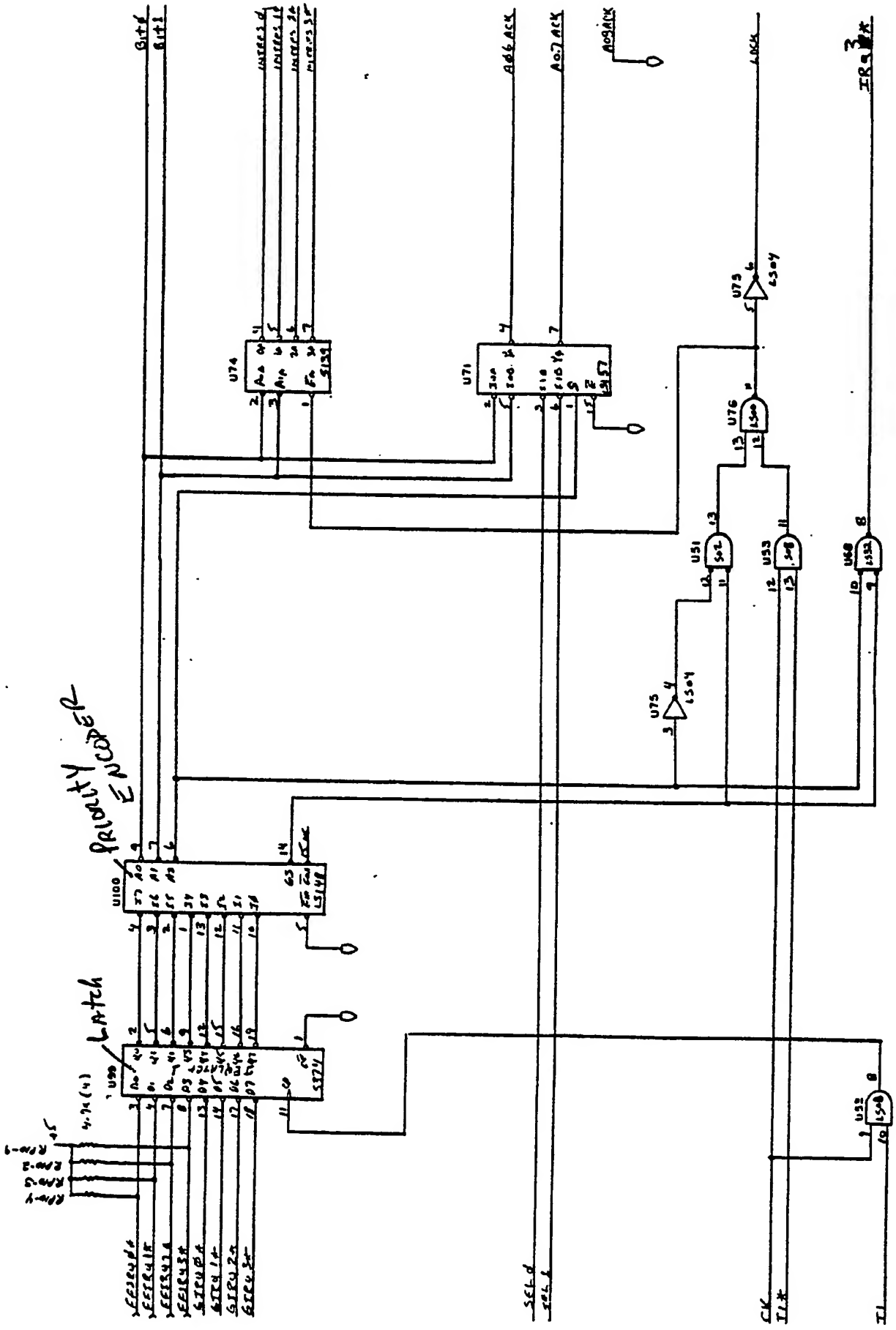
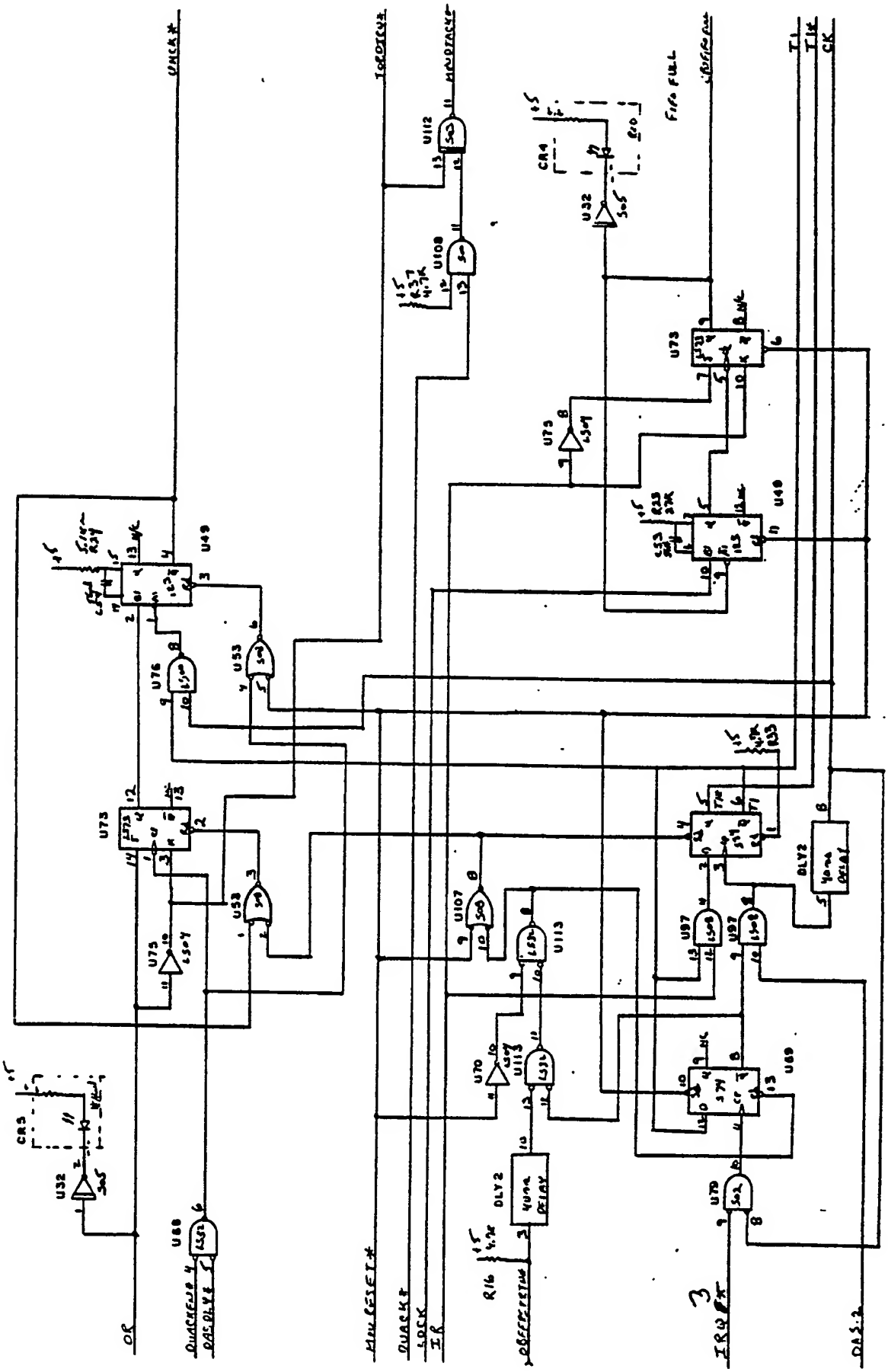

Priority Encoder
Latch
U74
U71
U75
U76
U51
U53
U68
U52
U100
U99
A06ACK
A07ACK
IRQ3

FIG. 26

29/41



FIG, 27

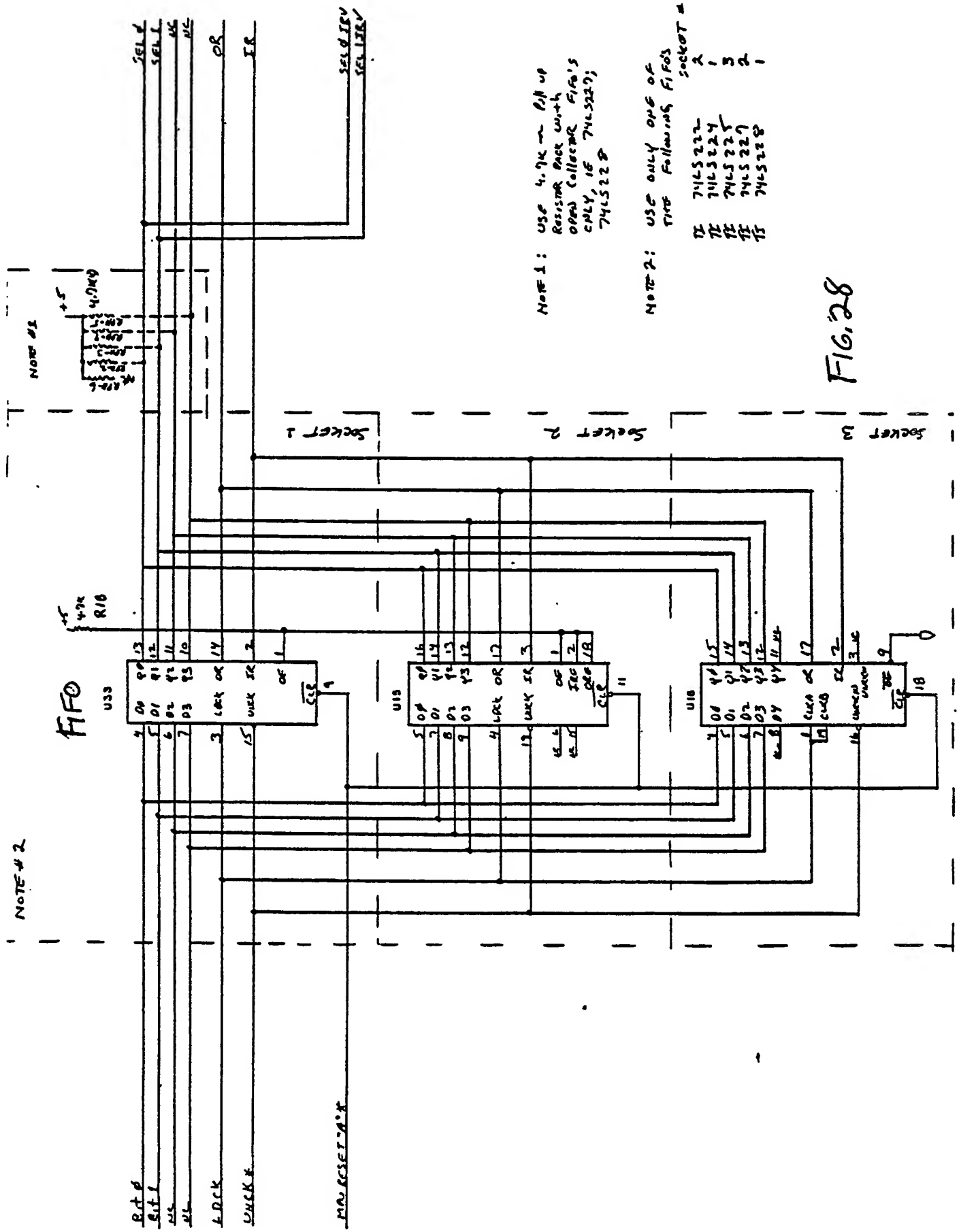

NOTE #1: USE 4.7K ~ 10K UP RESISTOR PACK WITH OPEN COLLECTOR FIFO'S ONLY, IE 74LS224
NOTE 2: USE ONLY ONE OF THE FOLLOWING FIFOS
socket 1
socket 2
socket 3
FIFO
U33
U15
U16
NOTE #2
LOCK
UNLOCK
MASTER RESET

FIG. 28

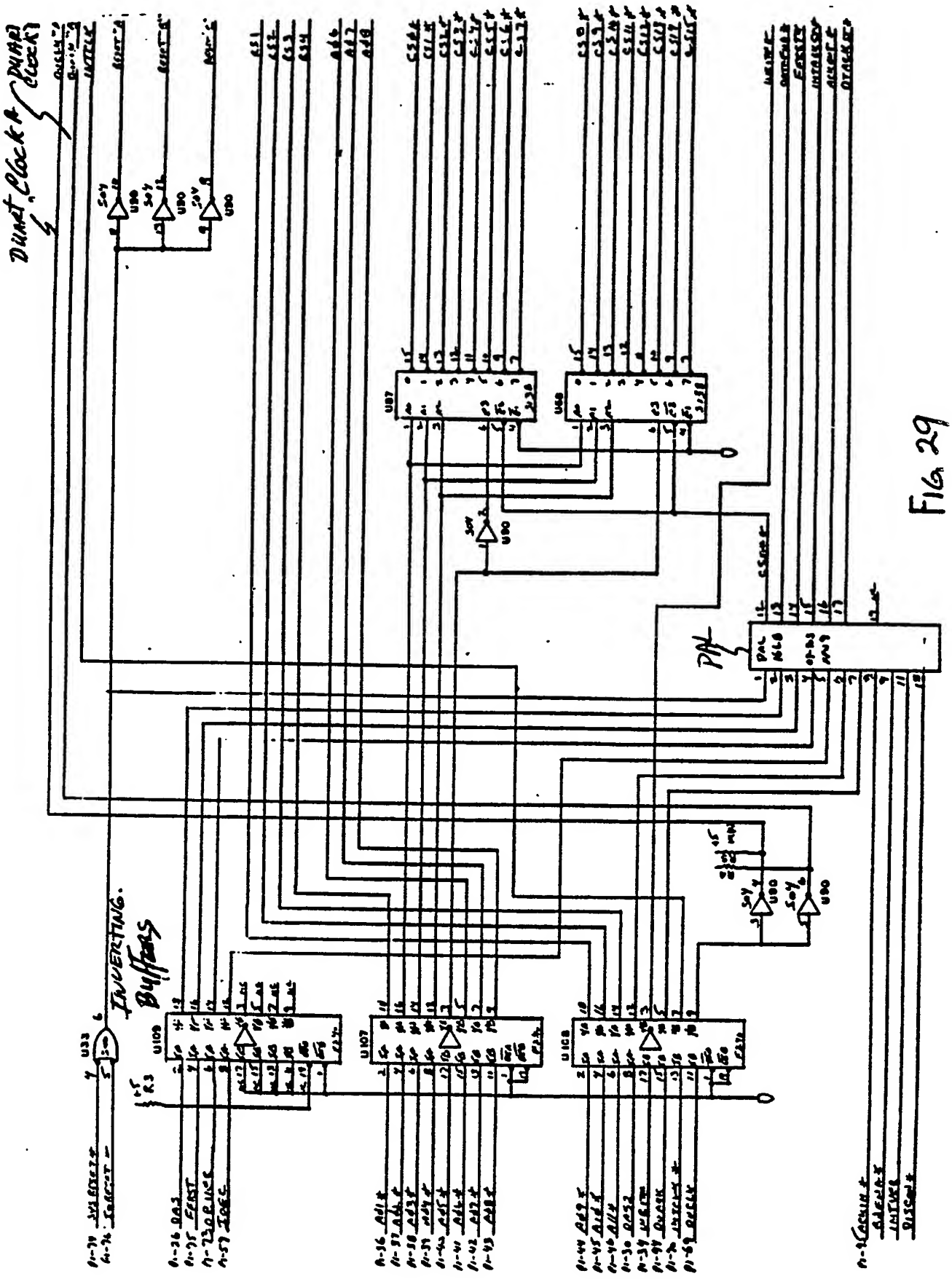


Fig 29

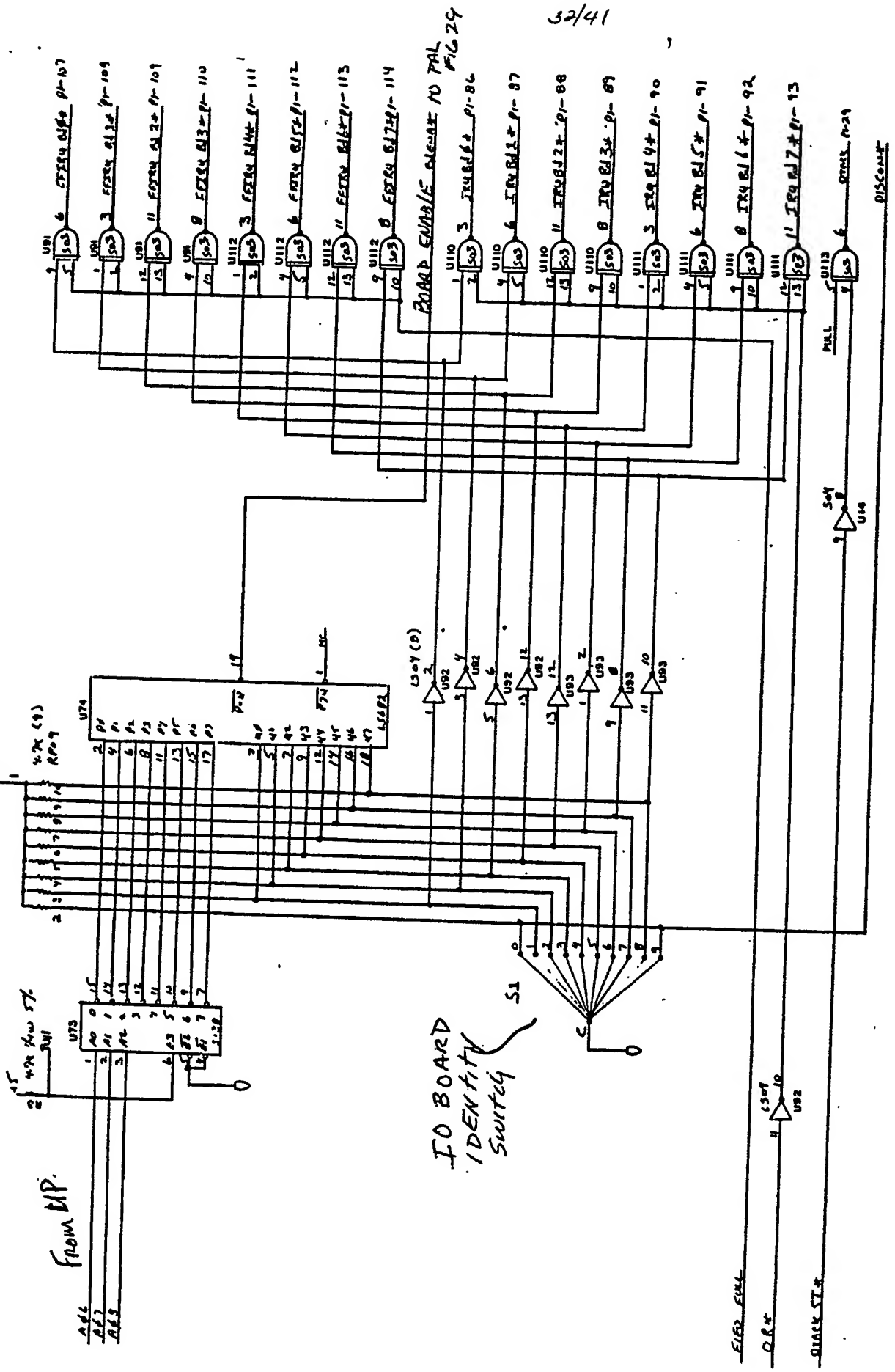
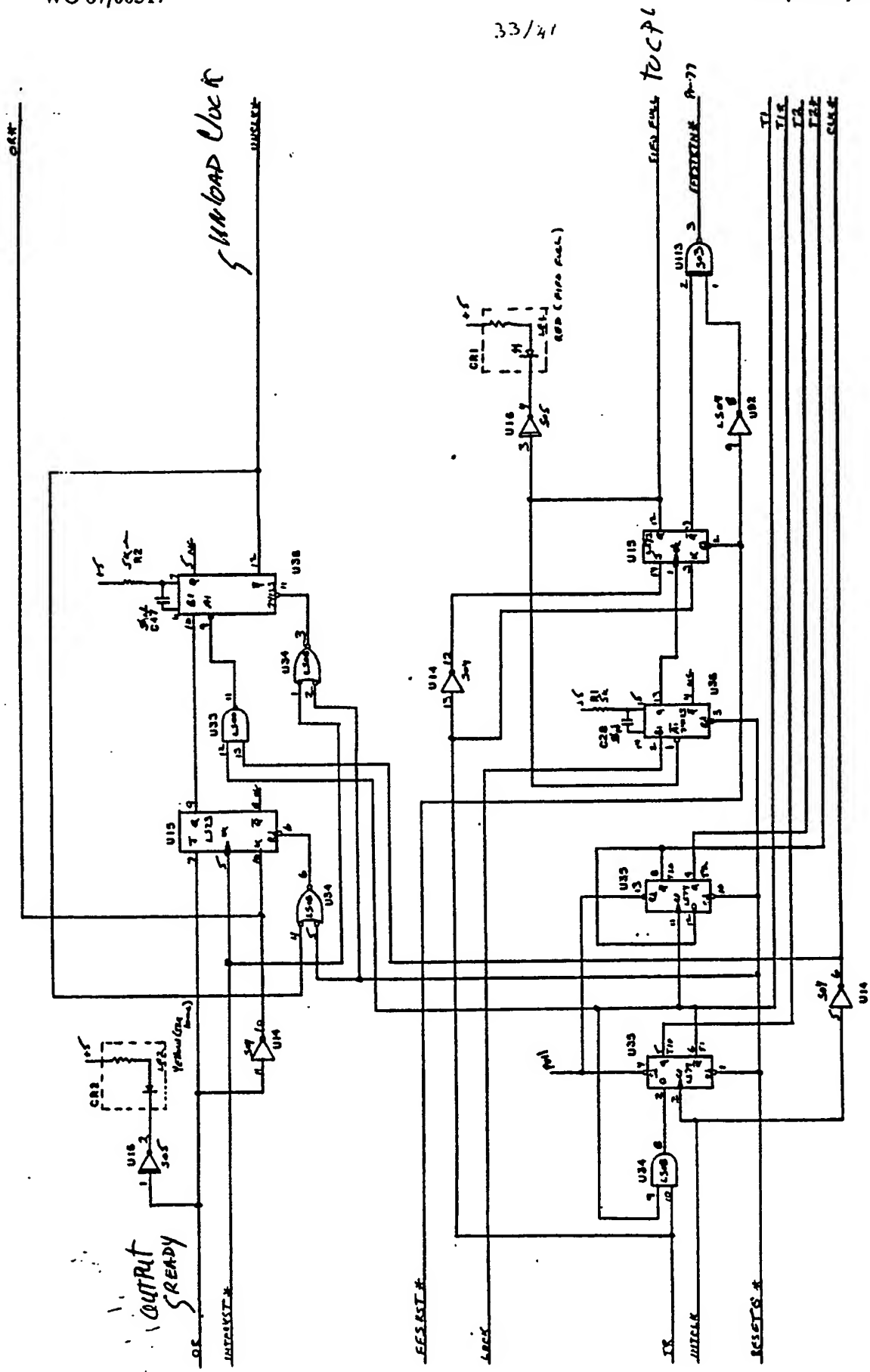

FROM BP
TO BOARD IDENTITY SWITCH
S1
BOARD ENABLE SIGNAL TO PAL FIG 29
DISCONNECT

FIG. 30



FIG, 31

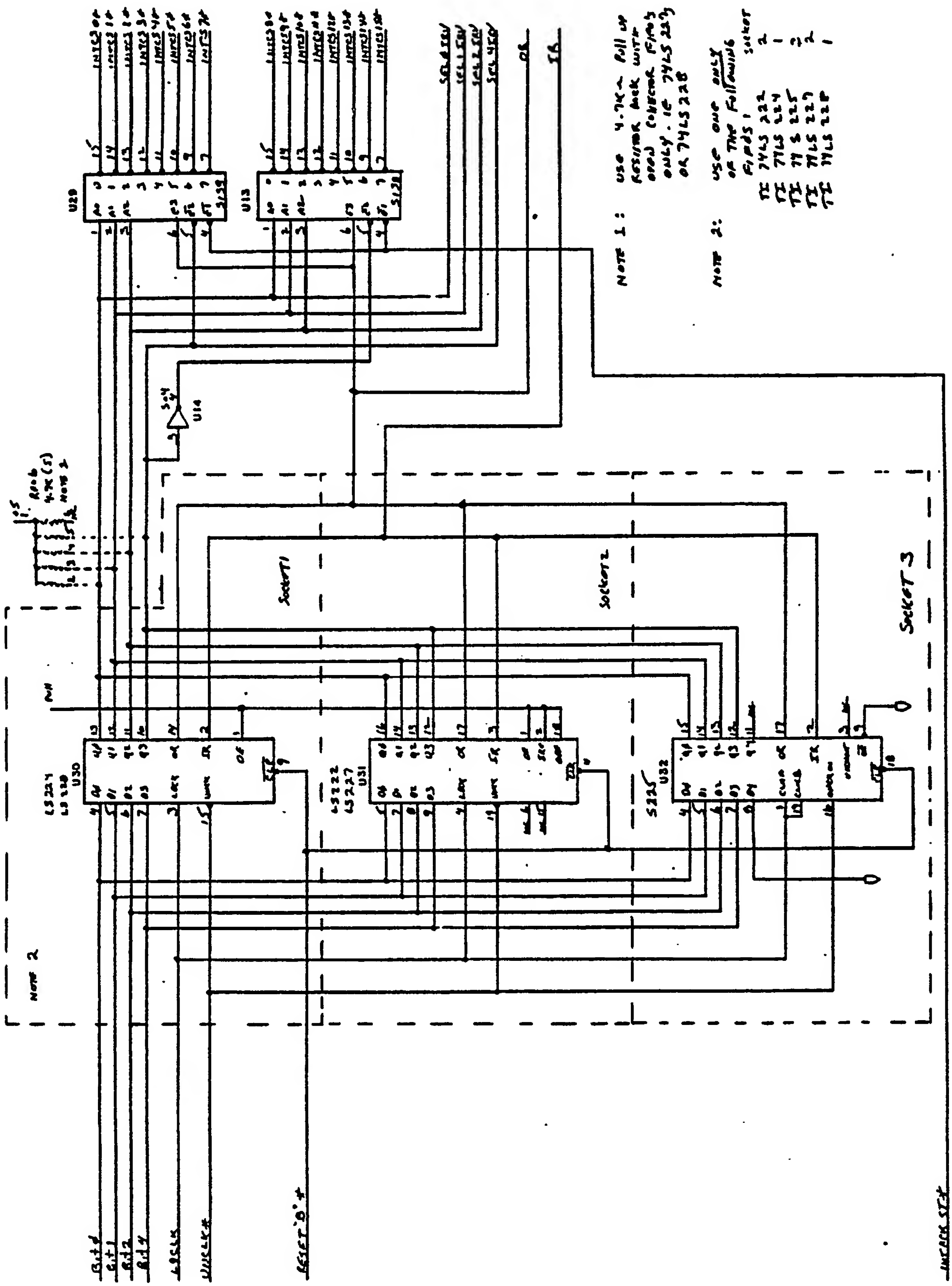

U30
U31
U32
U13
U29
U14
Socket 1
Socket 2
Socket 3
Note 1:
Note 2:
Note 2

FIG. 32

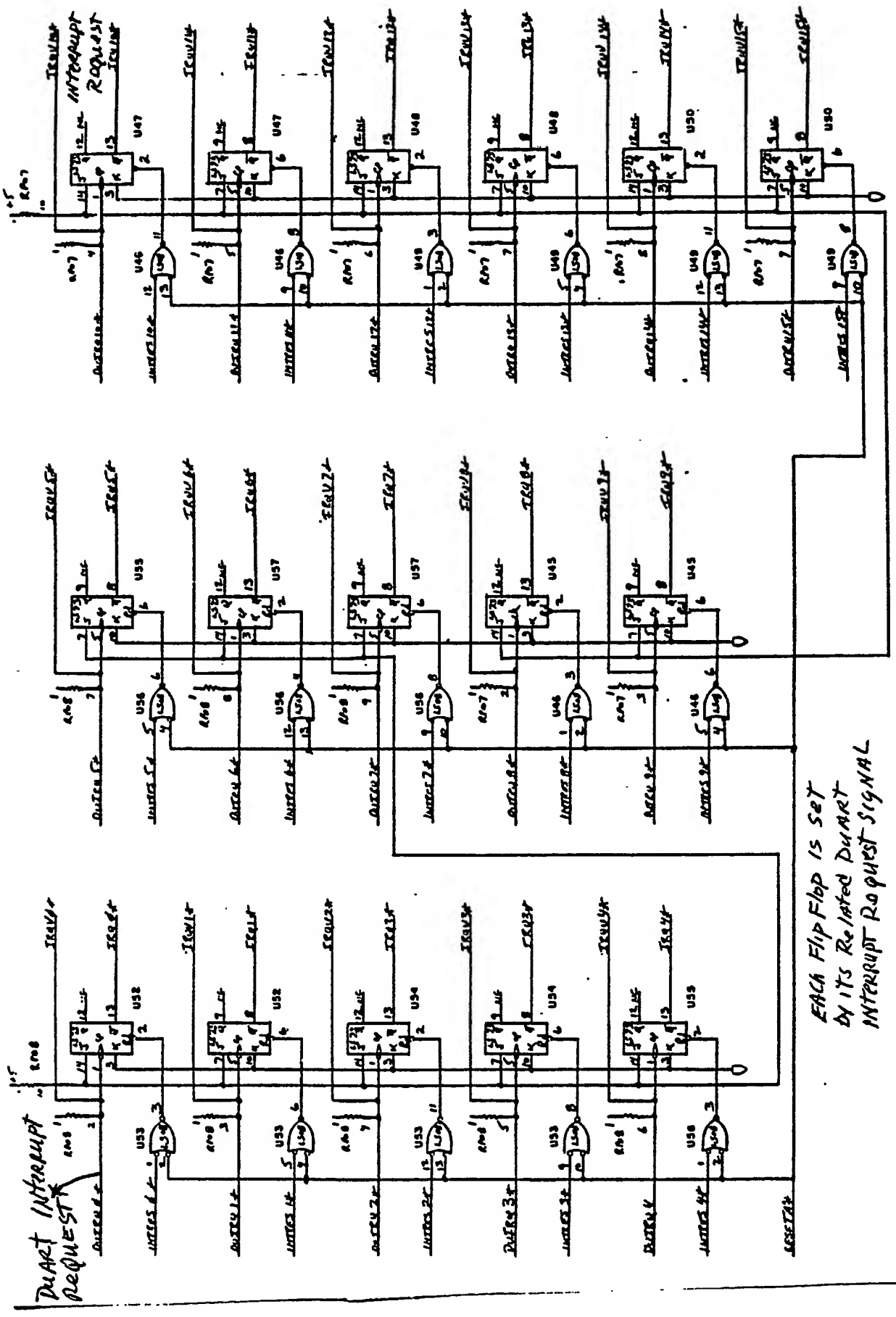

DUART INTERRUPT REQUEST
EACH FLIP FLOP IS SET BY ITS RELATED DUART INTERRUPT REQUEST SIGNAL

FIG. 33

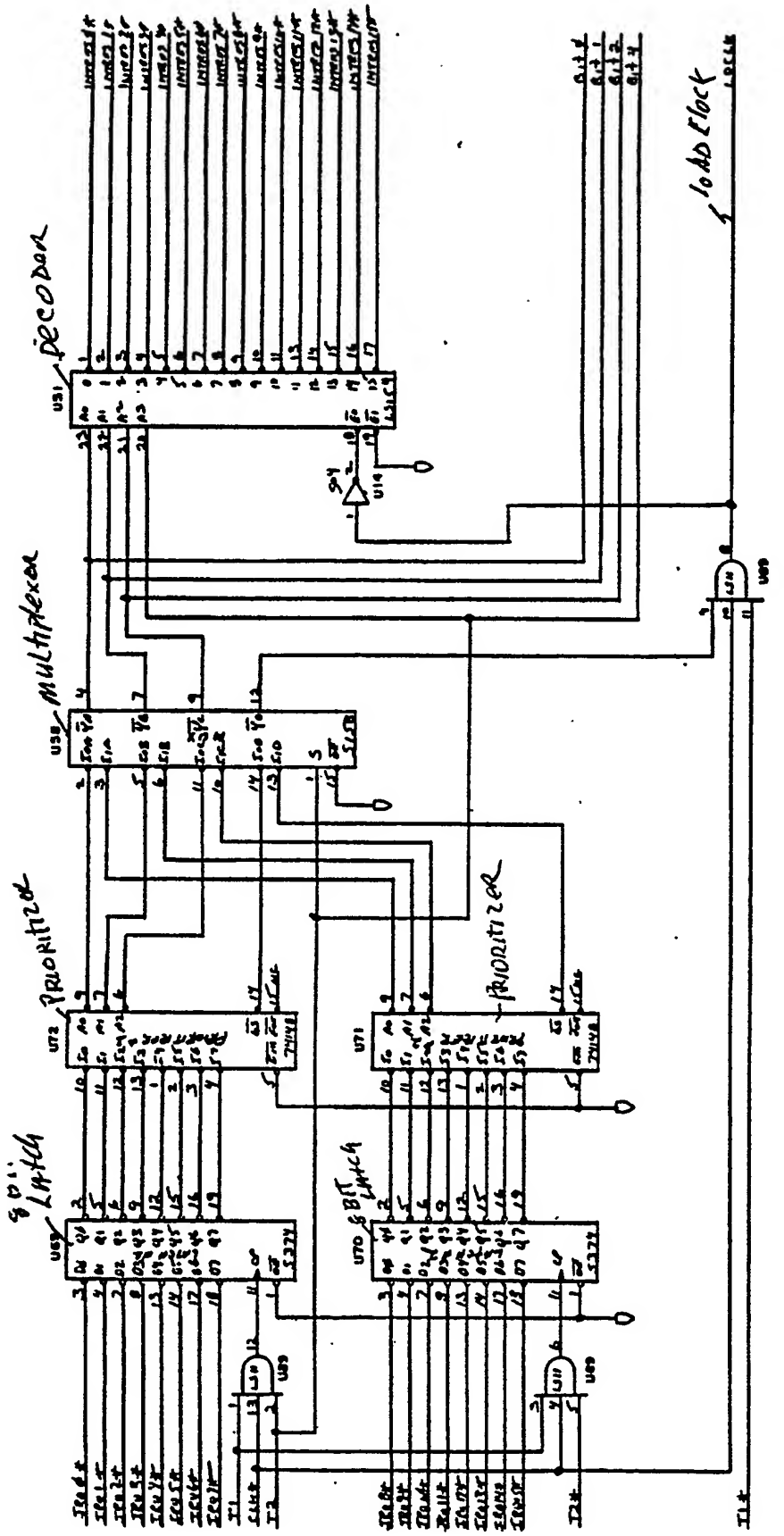

DECODER
MULTIPLEXER
PRIORITIZER
PRIORITIZER
9 BIT LATCH
9 BIT LATCH
LOAD CLOCK

FIG. 34

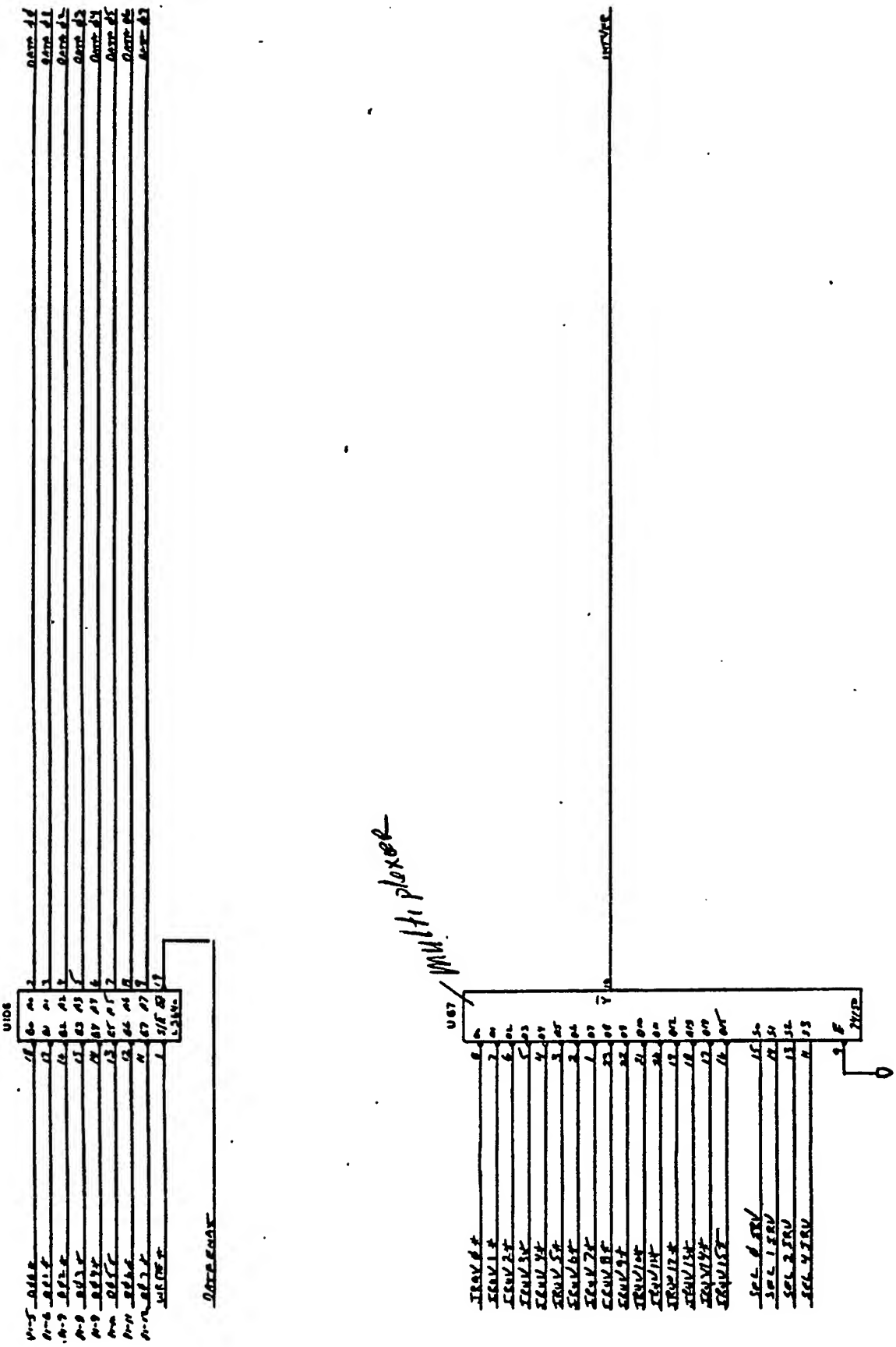


FIG. 35

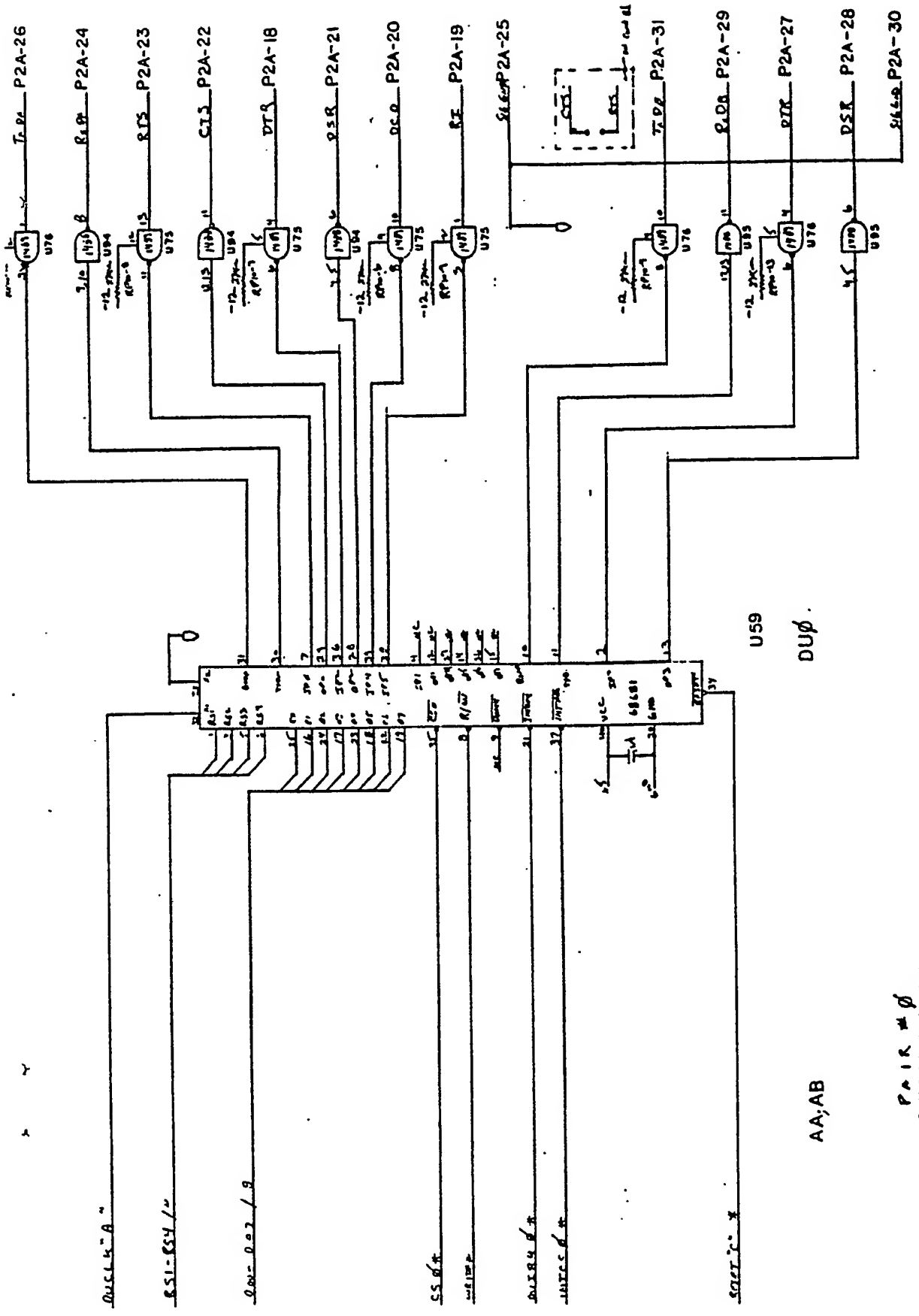

P2A-26
P2A-24
P2A-23
P2A-22
P2A-18
P2A-21
P2A-20
P2A-19
P2A-25
P2A-31
P2A-29
P2A-27
P2A-28
P2A-30
U59
DUØ
AA;AB

FIG. 36

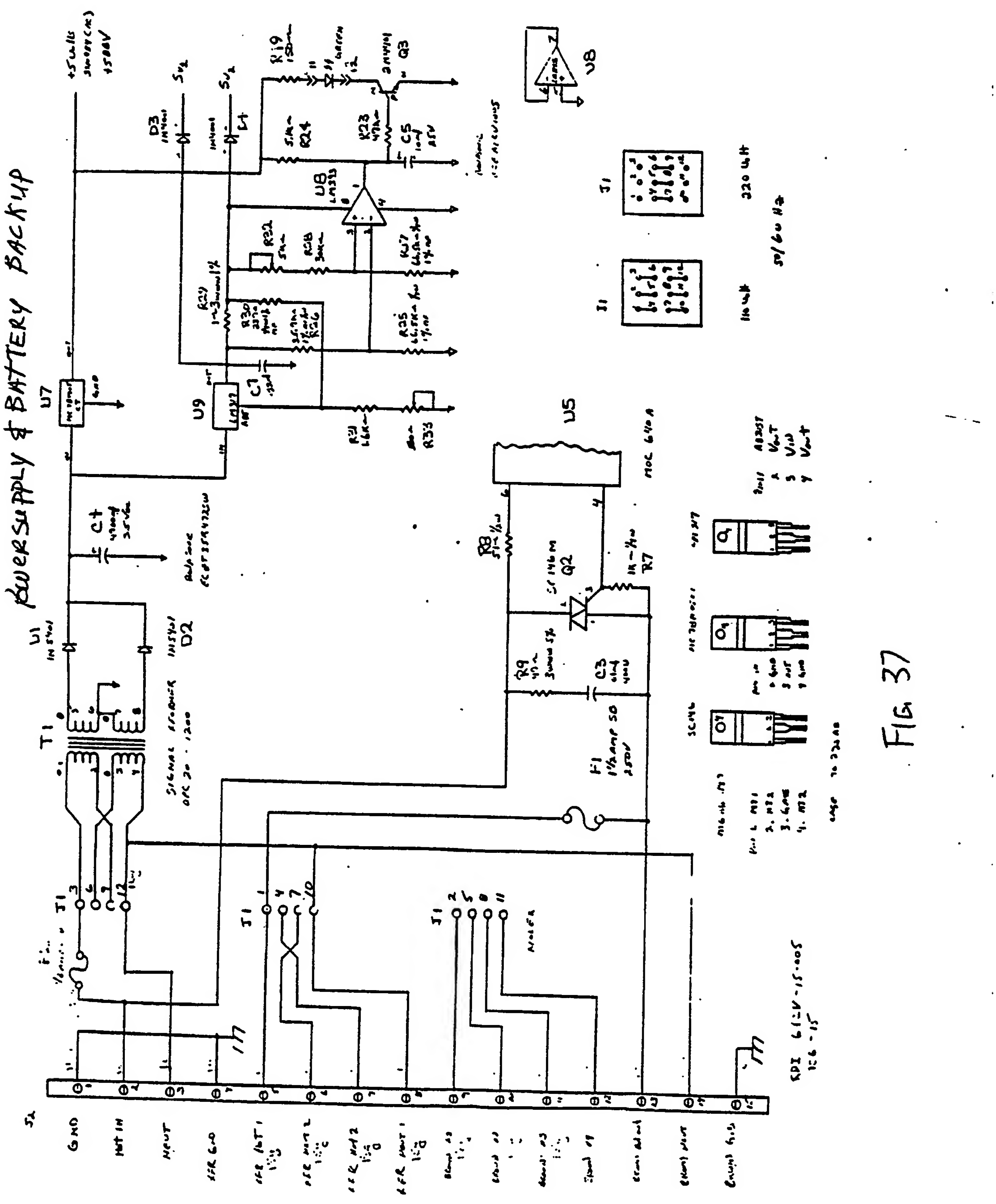

POWERSUPPLY & BATTERY BACKUP
T1
U1
D2
U7
U9
C7
U8
D3
R19
R24
R23
C5
Q3
U5
Q2
R7
R8
R9
C3
F1
R31
R33
R35
R26
R30
R32
R36
R37
LM393
MOC 640A
SC146M
GND
HOT IN
HOUT

FIG 37

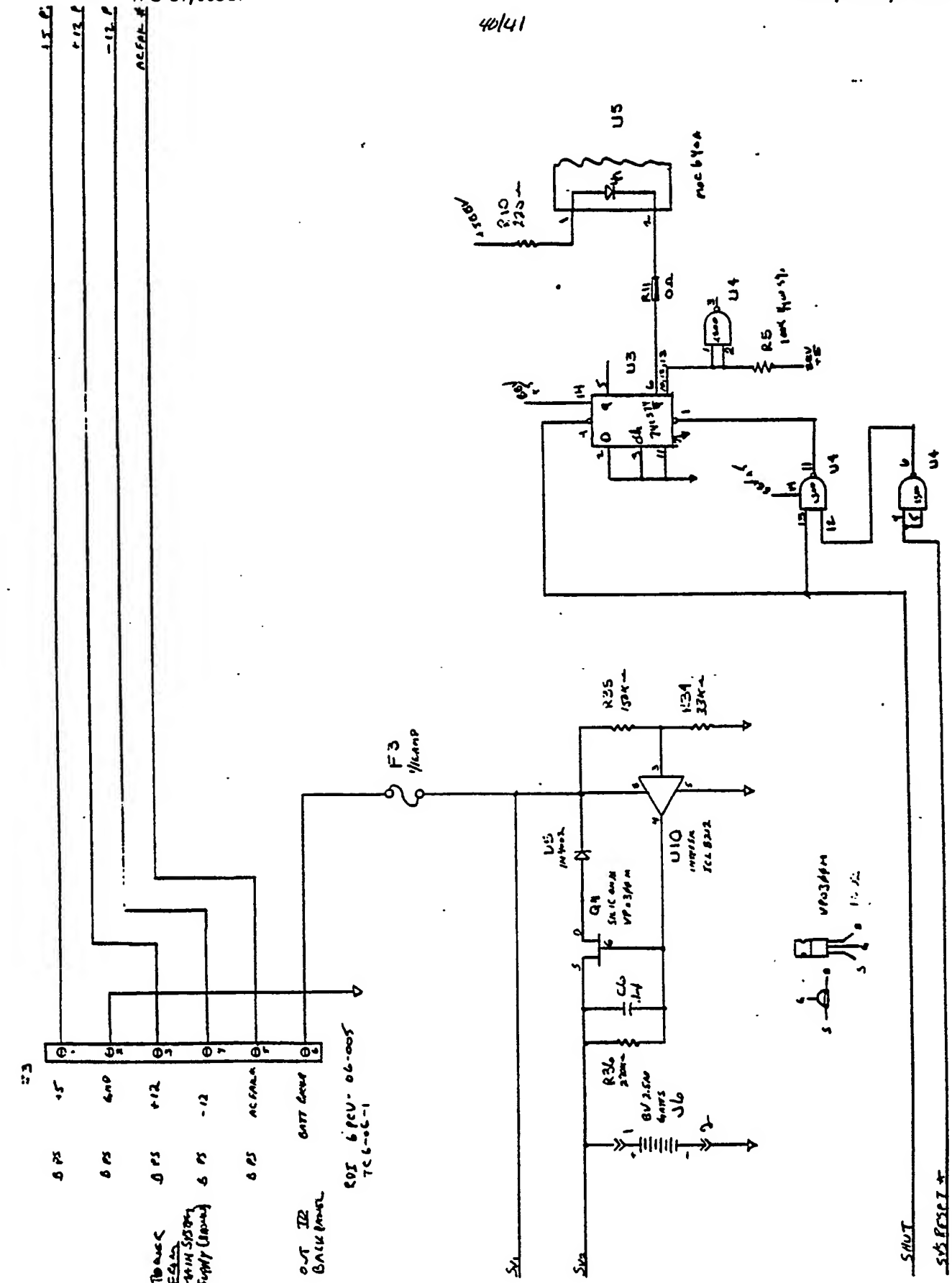


FIG. 38

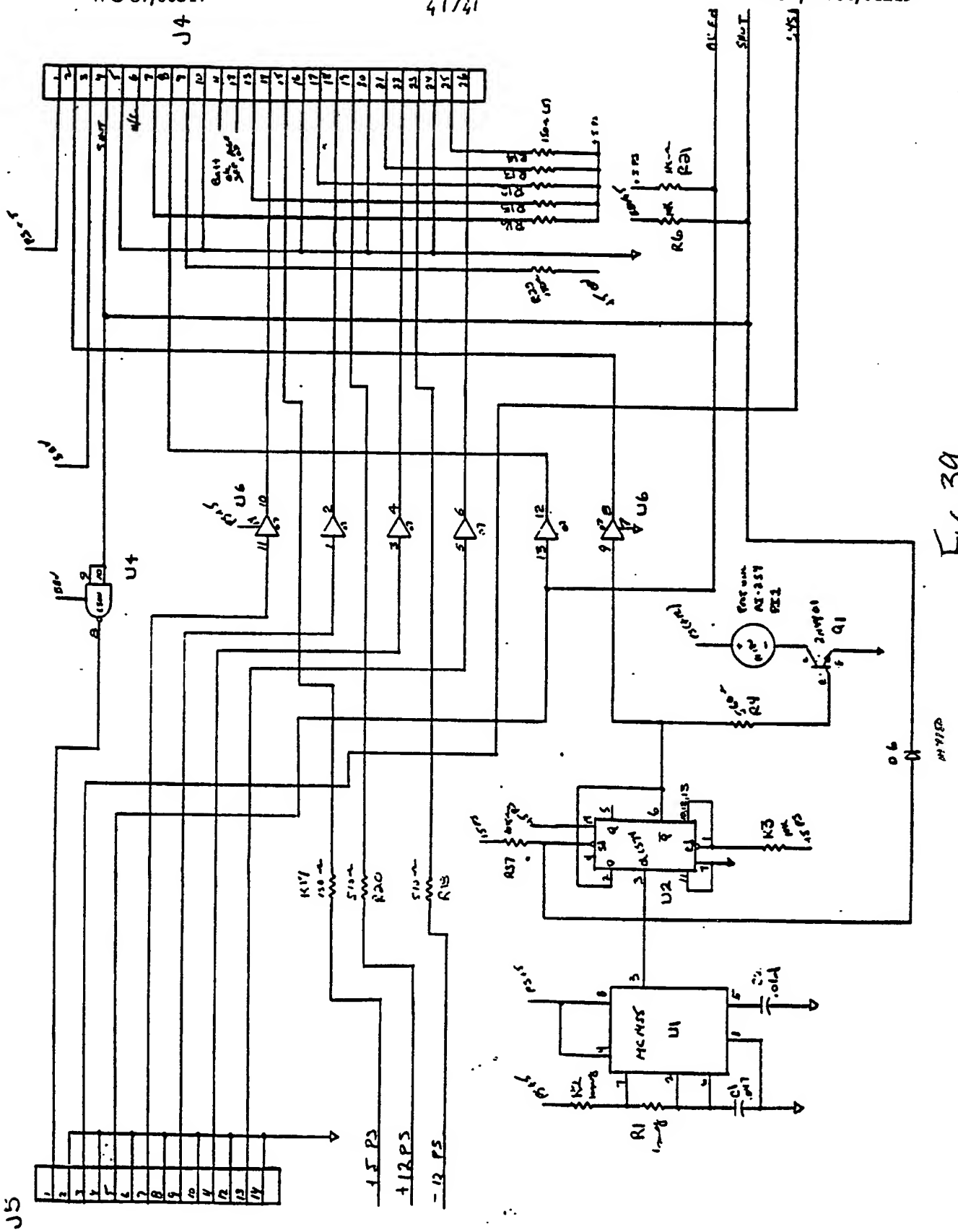

J4
J5
U1
U2
U4
U6
+5 P5
+12P5
-12 P5
SHUT D
SYSRESET
AC FAIL
STATUS 3*
STATUS 4*
STATUS 2*
STATUS 1*

FIG. 39

INTERNATIONAL SEARCH REPORT

International Application No PCT/US86/01219

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC (4): G06F 13/00, 13/14, 13/30
U.S. CL: 364/200, 900; 340/825.07

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
U.S.CL.	364/200, 900; 340/825.07

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [5]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
X	US, A, 4,034,346 (HOSTEIN) 5 July 1977 See the entire document.	1-3, 6-7
X	US, A, 4,124,889 KAUFMAN) 7 Nov. 1978 See Figures 1 and 5.	1-3, 6-7
A	US, A, 4,281,315 (BAUER) 28 July 1981 See Figures 1 and 2.	1-3, 6-7
X	US, A, 4,387,440(EATON) 7 June 1983 See Abstract.	3 and 4
A	US, A, 4,425,616 (WOODELL) 10 Jan. 1984 See Figure 1.	1-3, 6-7

* Special categories of cited documents: [15]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search [1]	Date of Mailing of this International Search Report [2]
1 AUG. 1986	20 AUG 1986
International Searching Authority [1]	**Signature of Authorized Officer [20]**
ISA/US	DAVID Y. ENG

Form PCT/ISA/210 (second sheet) (October 1981)